SIGNAL PROCESSING AND THE GLOBAL POSITIONING SYSTEM: THREE APPLICATIONS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Brady Whitson O'Hanlon

January 2017

SIGNAL PROCESSING AND THE GLOBAL POSITIONING SYSTEM: THREE APPLICATIONS

Brady Whitson O'Hanlon, Ph.D.

Cornell University 2017

Three research efforts are presented which develop new signal processing techniques as applied to the Global Positioning System. These three efforts are described in chapters that constitute stand-alone research papers. The first paper describes a software-defined radio that processes GPS signals, as well as a custom-built hardware platform that the software runs on. This receiver implements several novel processing techniques and was the result of collaborative work between The University of Texas at Austin, ASTRA, and Cornell University. The second paper discusses anomalies in the carrier phase of a particular GPS satellite discovered using the aforementioned software receiver. The final paper discusses a real-time implementation of a GPS spoofing detection method created as an addition to the software receiver, with detection results from a laboratory spoofing attack.

BIOGRAPHICAL SKETCH

Brady Whitson O'Hanlon was born in Rochester, New York on November 27, 1977. He moved to Ithaca, New York with his mother and sister at the age of four. He attended Ithaca public schools for 12 years, graduating from Ithaca High School. After an academic hiatus of several years, he attended Cornell University where he received a Bachelor of Science degree, majoring in Electrical and Computer Engineering. In the summers between his undergraduate studies he participated in several undergraduate research projects under the direction of Professor Michael Kelley, which served to pique his interest in space physics and in the tools used for ionospheric study. In the fall of 2007 he began a M.S./Ph.D. program in Electrical and Computer Engineering under the guidance of Professor Paul M. Kintner, Jr. After Professor Kintner passed away in the fall of 2010, Professor Mark L. Psiaki was kind enough to guide Brady through the remaining years of his graduate study. During graduate school he had the opportunity to teach the Electrical and Computer Engineering department's GPS course twice. He was fortunate to have participated in many field campaigns, travelling to such varied locales as Brazil, Peru, and Alaska. Brady successfully defended his thesis in December, 2016.

For my incredibly patient wife Kim and my loving parents Sally and Jim. It's been a long and interesting road, and they lit my way.

ACKNOWLEDGEMENTS

Many thanks to my co-advisers, Professors Paul M. Kintner, Jr. and Mark L. Psiaki. Paul and Mark were perfectly complementary, and I couldn't have done this without both of them. Dr. Todd Humphreys was an invaluable teacher and friend, and a daily inspiration. Many thanks as well to all of the contributors to the CASES receiver, both at the University of Texas at Austin and at Cornell.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . vi
List of Figures . . . viii
List of Tables . . . ix

1 Introduction 1

2 CASES: a Smart, Compact GPS Software Receiver for Space Weather Monitoring 3
2.1 Abstract . . . 3
2.2 Introduction . . . 4
2.3 Hardware Platform . . . 5
2.4 Novel Signal Processing Techniques . . . 8
2.4.1 Removal of local clock effects . . . 9
2.4.2 An advanced triggering mechanism for determining the onset of scintillation . . . 14
2.4.3 Data Buffering . . . 17
2.4.4 Data Bit Prediction . . . 18
2.5 Receiver Performance Analysis . . . 20
2.5.1 Measurement Precision . . . 20
2.5.2 Scintillation Robustness . . . 22
2.5.3 Comparison With a Commercial Scintillation Monitor . . . 25
2.6 Conclusions . . . 26

3 GPS Satellite Anomalies 28
3.1 Introduction . . . 28
3.2 Observations . . . 29
3.3 Conclusion . . . 32

4 Real-Time GPS Spoofing Detection via Correlation of Encrypted Signals 33
4.1 Abstract . . . 33
4.2 Introduction . . . 34
4.3 Architecture . . . 36
4.4 Signal Model and Pre-Processing . . . 39
4.4.1 Signal Model . . . 39
4.4.2 Tracking the C/A Signal . . . 40
4.4.3 Received Signal Power . . . 42
4.5 Methodology . . . 44
4.5.1 Spoofing Detection Statistic Calculation . . . 44

4.5.2 Spoofing Detection Hypothesis Test 53
4.6 Results 56
4.6.1 Determining The Per-Satellite Power Variation 56
4.6.2 Detection of a Spoofing Attack 58
4.7 Summary and Conclusion 61

5 Summary and Conclusions **65**

Bibliography **67**

LIST OF FIGURES

2.1 Receiver hardware block diagram. 6
2.2 CASES in two different form factors. 7
2.3 De-trended beat carrier phase for two satellites. 12
2.4 Amplitude and phase scintillation of a GPS signal. 13
2.5 Power spectrum of the complex channel response functions of a scintillating signal. Frequency bands used for the scintillation power ratio are shown in orange. 15
2.6 Illustration of the benefits of data buffering. 18
2.7 Single-receiver dual frequency ionospheric delay at L1 (bottom two panels) and inter-receiver ionospheric delay difference (top panel). Code-derived values are in blue and carrier-phase-derived values are in red. The receivers used a common antenna, and the carrier-phase-derived data have had a bias relative to the code-derived data removed. 21
2.8 Single-receiver dual frequency ionospheric delay at L1 (bottom two panels) and inter-receiver ionospheric delay difference (top panel). Code-derived values are in blue and carrier-phase-derived values are in red. The receivers used independent antennas. 23
2.9 Phase error without use of data bit prediction. 24
2.10 Phase error with data bit prediction. 25
2.11 Amplitude scintillation observed by CASES and a commercial scintillation monitor. 26
2.12 Detail of amplitude scintillation observed by CASES and a commercial scintillation monitor. 27

3.1 Square pulses on phase-based TEC due to L1 C/A carrier phase anomalies. 29
3.2 A typical anomaly as observed in the (negative) range-equivalent L1 C/A carrier phase 31

4.1 An example receiver architecture. 38
4.2 Filtered P(Y) code normalized autocorrelation function. 50
4.3 Normalized spoofing detection statistic and related quantities for a spoofed signal (top panel) and an un-spoofed signal (bottom panel) during a spoofing attack. 59
4.4 Receiver output for all signals during a spoofing attack. 61
4.5 Probability of detection for all signals during a spoofing attack. . 62

LIST OF TABLES

4.1 Observed decrement in transmitted power between L1 C/A and L1 P(Y) signals, by satellite. 64

CHAPTER 1

INTRODUCTION

Use of radionavigation satellite systems such as the U.S. Global Positioning System has become ubiquitous for its intended purposes of navigation and timing. The fact that the signals from these systems traverse the atmosphere en route to terrestrial users has made them an invaluable remote sensing tool for the study of the troposphere and ionosphere. Commercial receivers are often inadequate for the unique requirements of atmospheric scientists because the receivers either do not produce appropriate measurements or the manner in which the measurements are produced are unknown to the users due to the generally proprietary nature of commercial products.

In the interest of producing a GPS receiver uniquely suited to ionospheric study, a software-defined receiver was built: the Connected Autonomous Space Environment Sensor (CASES). This receiver was designed to be inexpensive, easy to modify via software changes, and capable of producing measurements both useful to and understandable by ionospheric scientists. The design and features of this receiver, both the software and the custom-designed hardware upon which it runs are described in detail in Chapter 2.

In the course of using this receiver to study variations in Total Electron Content (TEC), it was discovered that the phase of the carrier signal from one particular satellite was behaving in a manner not described by the civilian GPS interface specification [1]. Specifically, the carrier was exhibiting aperiodic step changes on the order of 10 degrees. These newly discovered phase anomalies are discussed in Chapter 3.

The open nature of civilian GPS signals makes those signals vulnerable to spoofing [2, 3, 4, 5], the transmission of signals intended to appear as legitimate GPS signals for the purpose of deceiving users of those signals. One defense against GPS signal spoofing involves making use of the unknown but presumably secure (and thus un-spoofable) P(Y) code [6, 7, 8]. To demonstrate this method in real-time, the aforementioned software-defined GPS receiver CASES was modified to implement this method. The details of this implementation and experimental results therefrom are described in Chapter 4.

CHAPTER 2

CASES: A SMART, COMPACT GPS SOFTWARE RECEIVER FOR SPACE WEATHER MONITORING

B.W. OHanlon, M.L. Psiaki, S. Powell, J.A. Bhatti, T.E. Humphreys, G. Crowley, Geoff and G. Bust. "CASES: A smart, compact GPS software receiver for space weather monitoring," *Proceedings of the ION GNSS Meeting*, 2011.

2.1 Abstract

A real-time software-defined GPS receiver for the L1 C/A and L2C codes has been developed as a low-cost space weather instrument for monitoring ionospheric scintillation and total electron content. The so-called CASES receiver implements several novel signal processing techniques not previously published that make it well suited for space weather monitoring: (A) a differencing technique for eliminating local clock effects, (B) an advanced triggering mechanism for determining the onset of scintillation, (C) data buffering to permit observation of the prelude to scintillation, and (D) data-bit prediction and wipe-off for robust tracking. The receiver has been tested in a variety of benign and adverse signal conditions (e.g., severe ionospheric scintillation, both real and simulated); the results are presented here. The custom hardware platform on which the software runs is compact while remaining flexible and extensible. The CASES platform consists of a digital signal processor, an ARM microcontroller, and a custom-built narrow-band dual-frequency front end. Because the receiver is software-defined, it can be remotely reprogrammed via the Internet

or another communications link.

2.2 Introduction

CASES (Connected Autonomous Space Environment Sensor) was designed to facilitate ionospheric study. Study of the Earth's ionosphere is a particularly difficult proposition due to its location, spanning from one hundred kilometers or so to greater than one thousand kilometers above the Earths surface. As the signals from GPS satellites traverse this region and are changed by disturbances therein, they provide a unique tool for studying the structure of the ionosphere and its variations. GPS signals are changed in two ways of particular interest: refraction due to the presence of charged particles in the ray path, and diffraction due to the occasionally irregular densities of those charged particles. The path-integrated number of electrons (total electron content, or TEC) can be observed by comparing observations on multiple frequencies. The effects of density irregularities manifest as rapid fluctuations of signal amplitude and/or phase (ionospheric amplitude and phase scintillation, respectively). GPS receivers have been used to study both of these effects for many years [9]. The CASES receiver differs from typical GNSS receivers in two key ways: it has been specially designed to measure TEC and scintillation parameters, and special features have been implemented that allow it to operate robustly in the presence of vigorous ionospheric scintillation. The estimation of TEC will be lightly treated here, as scintillation provides a much more challenging signal environment than any observed TEC fluctuations, and the measurements needed to estimate TEC are produced in the course of standard receiver operation (i.e., code and carrier phase measurements). Signal variations due to tropospheric effects are not

addressed here.

Section 2.3 of this paper contains a description of the CASES hardware platform, the available peripherals, and the performance specifications. Section 2.4 describes the various novel processing techniques implemented by the receiver. Section 2.5 contains an analysis of the receiver performance under various signal conditions, and Section 2.6 contains conclusions about the platform. CASES is the result of development effort between Cornell University, the University of Texas at Austin, and ASTRA [10, 11].

2.3 Hardware Platform

The CASES receiver was designed with the goal of providing a capable platform with many peripheral options while remaining inexpensive, relatively small, and power-efficient. The final configuration has three main components: a custom-built dual-frequency front end, a digital signal processor board, and a "single board computer" (SBC) featuring an ARM microcontroller. A block diagram of the receiver hardware is shown in Fig. 2.1, and a photograph of the receiver in two different configurations is shown in Fig. 2.2.

The front-end performs automatic gain-controlled amplification, filtering, mixing to intermediate frequency, and sampling. The front-end has a relatively narrow bandwidth of 2.4 MHz, and produces 2-bit samples at 5.7 MSamples/second. As it is a dual-frequency front-end, it produces one set of 2-bit samples for each of the GPS L1 and L2 frequencies. An on-board temperature-compensated crystal oscillator (TCXO) is the frequency reference for both frequencies, and both signals are sampled synchronously. Although the use of

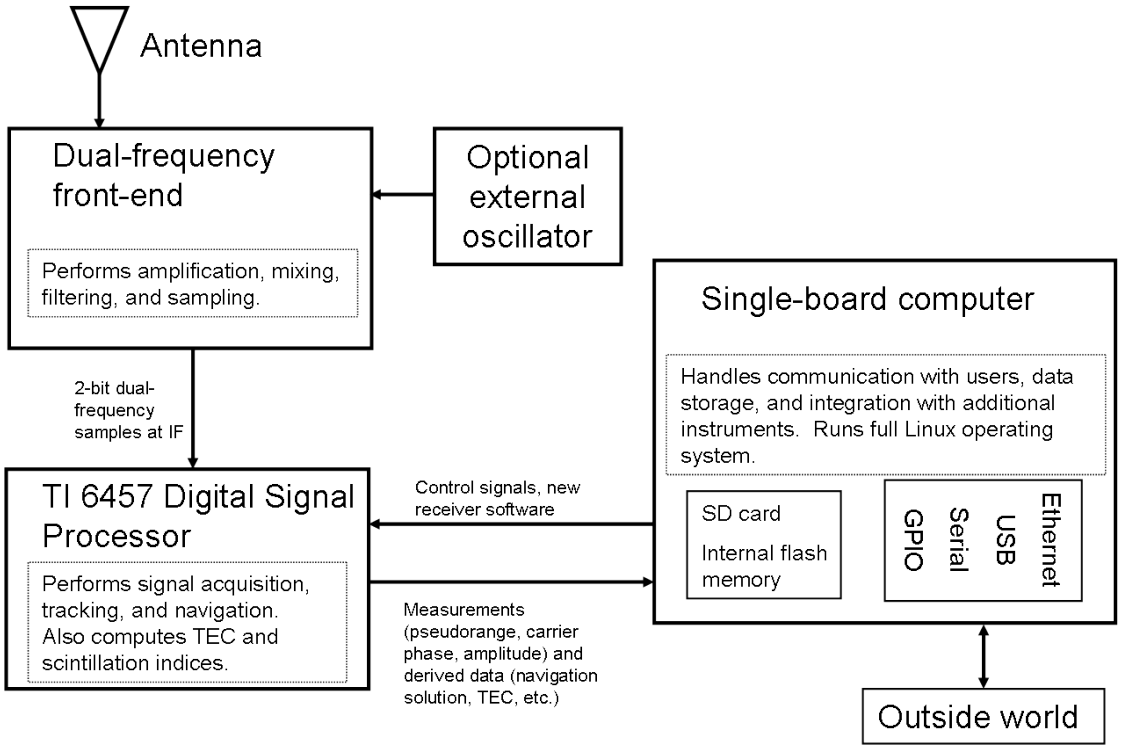

Figure 2.1: Receiver hardware block diagram.

a TCXO introduces non-negligible variations in measured carrier phase [12], a method to remove this error has been implemented, as discussed in Section 2.4. The front-end can provide a selectable 5 volt DC bias on the antenna input for powering active antennas, and has an optional input for connecting an external 10 MHz frequency reference, with termination of 50 or 1000 Ohms. The board consumes approximately 360 milliamps at 5 volts, excluding any power required by a connected active antenna.

CASES is a software-defined receiver, with all processing downstream of the front end performed on a general-purpose digital signal processor. A second custom-designed board houses a Texas Instruments C6457 digital signal processor (DSP). The processor runs at a 1 GHz clock speed, has 2 MB of on-chip RAM, 128 MB of off-chip RAM, and 4MB of non-volatile flash memory. The so-called DSP board performs all acquisition and tracking functions, as well as

Figure 2.2: CASES in two different form factors.

computation of the navigation solution and various observables such as pseudorange, beat carrier phase, and Doppler shift. The board outputs in-phase and quadrature accumulations, beat carrier phase, and timestamps at up to 100 Hz, and all other data at 10 Hz or less. Processor utilization while tracking 12 GPS L1 C/A code channels and 4 GPS L2CL channels as well as computing the navigation solution, performing continuous background signal acquisition, and all other overhead is roughly 75%. The DSP board consumes approximately 580 milliamps at 5 volts.

The third main receiver component is an SBC running the GNU/Linux operating system. The SBC features an ARM AT91SAM9260 microcontroller with a host of available peripherals. This board features 32 MB of RAM and 128 MB of

flash memory for the file system. Available peripherals include Ethernet, serial peripheral interface, a secure digital card reader, universal serial bus, ZigBee, Wi-Fi, a 10-bit analog-to-digital converter, and general-purpose I/O pins. Communication is typically done via RS232 serial port, Ethernet, or Wi-Fi. The SBC runs a network-connected server program that allows remote monitoring, data logging, and uploading of new code images or configuration files. Additionally, it runs a secure shell server to allow remote log-in for additional operations not provided by the server program. The SBC consumes approximately 260 milliamps at 5 volts.

2.4 Novel Signal Processing Techniques

Density irregularities in ionospheric plasma can induce rapid fluctuations in the phase and/or amplitude of GPS signals, which can cause the receiver to lose signal lock [9, 13]. This problem is even more pronounced for GPS L2P(Y) signals in receivers that employ codeless or semi-codeless tracking techniques, which are more prone to losing lock on the signal due to various losses introduced by the processing [14], and are not well suited for measuring phase scintillation on L2 due to the low tracking loop bandwidth they typically employ [13]. A quantitative relationship between scintillation effects on multiple frequencies is not well understood, though it has long been known that the correlation between multiple frequencies is inversely related to the scintillation intensity (see Fig. 1 of [15]). Thus, a dual-frequency receiver is desirable both for estimating TEC and for estimating ionospheric scintillation parameters at disparate frequencies, as multiple-frequency scintillation parameter estimation provides non-redundant information. Several techniques have been imple-

mented to make this receiver particularly well suited for scintillation monitoring.

2.4.1 Removal of local clock effects

Ionospheric scintillation severity is typically characterized by two parameters: S_4, the normalized signal amplitude standard deviation, and σ_ϕ, the carrier phase standard deviation [16]. Unfortunately, the phase noise introduced by a receiver's TCXO (such as the one used in CASES) is spectrally similar to the phase fluctuations from ionospheric scintillation (see Fig. 2 in [12]). To prevent local clock variations from contaminating estimated scintillation parameters, the CASES receiver pre-processes the carrier phase time histories to remove common-mode clock effects, prior to estimating the scintillation parameters. The key idea in this algorithm is that the TCXO-induced phase noise can be estimated by observing phase fluctuations from a signal that is known to be free of ionosphere-induced phase variations [17]. Just how one knows that a signal is free of these fluctuations prior to the calculation of scintillation parameters whose validity depends on this assumption is a bit of a chicken-and-egg conundrum, but it is readily resolvable, as described in subsection 2.4.2 below. For now, let us assume that a suitable reference signal free of ionospheric scintillation has been identified. The clock effect removal algorithm starts by modeling the beat carrier phase measurements from the n^{th} tracking channel as

$$\phi_n = \phi_{n-geom} + \phi_{clk} + \phi_{n-scint} + n_n \tag{2.1}$$

where ϕ_{n-geom} is the phase component due to satellite geometry, ϕ_{clk} is the phase

component due to oscillator noise, $\phi_{n-scint}$ is the phase component due to ionospheric effects, and n_n is other noise sources (e.g., thermal noise, satellite oscillator phase noise, multipath). Assume that a reference channel free of ionospheric effects has been identified. The phase of the reference channel is modeled as

$$\phi_{ref} = \phi_{ref-geom} + \phi_{clk} + n_{ref} \tag{2.2}$$

where the same notation applies as previously, but for the reference channel rather than the n^{th} tracking channel. The difference of these carrier phases is taken, creating a combined carrier phase measurement given by

$$\begin{aligned} \tilde{\phi}_n &= \phi_n - \phi_{ref} \\ &= \phi_{n-geom} - \phi_{ref-geom} + \phi_{n-scint} + n_n + n_{ref} \end{aligned} \tag{2.3}$$

This new phase measurement now contains the combination of the geometric effects for the two channels, the combination of the noise on the two channels, and the phase fluctuations due to scintillation on channel n.

In the next processing step, the differential phase due to geometric effects $\Delta_{geom} \equiv \phi_{n-geom} - \phi_{ref-geom}$ is removed. Over time intervals of up to 100 seconds, and for stationary GPS receivers, Δ_{geom} can be accurately modeled as a 3^{rd} order polynomial. It can then be removed by subtracting the 3^{rd} order polynomial fit of Δ_{geom} over a 100 second interval from the combined carrier phase measurement. This procedure removes the Δ_{geom} component while leaving $\phi_{n-scint}$ unaffected at the frequencies of interest (greater than about 0.2 Hz). The resulting phase after removal of differential geometry terms is modeled as

$$\tilde{\phi}_{nf} = \phi_{n-scint} + n_n - n_{ref} \tag{2.4}$$

Thus, the phase scintillation on channel n is isolated from local clock and satellite motion effects. It should be noted, however, that $\tilde{\phi}_{nf}$ is a filtered version of the phase scintillation effects on channel n. Given that vigorous phase scintillation often contains substantial power well beyond the bandwidth of a typical phase tracking loop (e.g., beyond 10 Hz) [17], high-frequency scintillation effects are not present in $\tilde{\phi}_{nf}$. To recover the high-frequency variations induced by scintillation – up to the pre-detection bandwidth $B_{pd} = 1/T_a$, where T_a is the correlation accumulation interval – the instantaneous phase angle of the in-phase and quadrature accumulations $\phi_{IQn} = atan2(Q_n \cdot d, I_n \cdot d)$ is added to $\tilde{\phi}_{nf}$

$$\tilde{\phi}_{npd} = \tilde{\phi}_{nf} + \phi_{IQn} \tag{2.5}$$

Here, d is the ± 1 valued navigation data bit that was in effect over the interval corresponding to I_n and Q_n. The quantity $\tilde{\phi}_{npd}$ includes all scintillation frequencies up to the pre-detection bandwidth. For typical $T_a = 0.02s$, $B_{pd} = 50Hz$, which is sufficient to capture even vigorous phase scintillation.

Alternatively, the complex channel response function [18] $z(t)$ can be produced simply by rotating the vector defined by the I_n and Q_n accumulation values by the phase $\tilde{\phi}_{nf}$.

The penalty paid for adding ϕ_{IQn} to $\tilde{\phi}_{nf}$ is, of course, that ϕ_{IQn} includes high-frequency noise in addition to possible high-frequency scintillation. Despite this, $\tilde{\phi}_{npd}$ is a useful new quantity for study of phase scintillation because it is

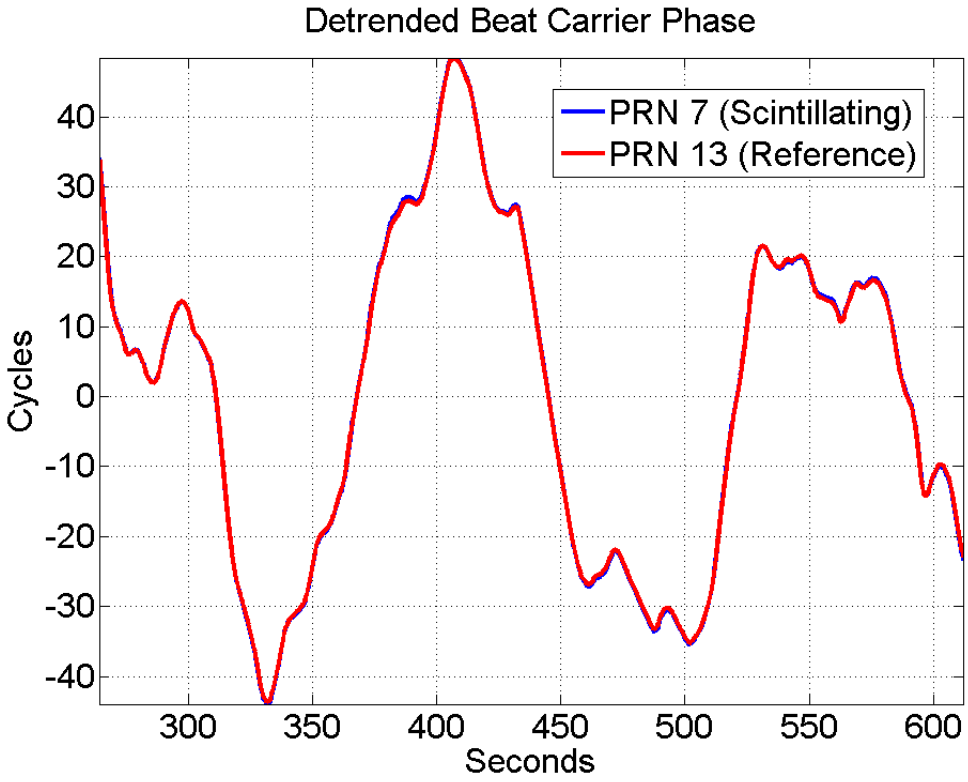

Figure 2.3: De-trended beat carrier phase for two satellites.

free of local clock, satellite geometry, and phase tracking loop effects.

To illustrate the effectiveness of phase pre-processing techniques, consider Figs. 2.3 and 2.4. Fig. 2.3 shows data gathered by the author during a scintillation campaign at the Jicamarca Radio Observatory near Lima, Peru, in March of 2011. The red signal was the reference channel, and the blue signal (barely visible underneath the red trace) was strongly scintillating.

What is shown is simply the beat carrier phase of the two signals after fitting and removal of a 3^{rd} order polynomial to eliminate geometric effects. The two lines are virtually indistinguishable, indicating that the majority of variation is due to clock effects. Fig. 2.4 shows the carrier phase (in green) of this same scintillating signal, after removal of local clock effects and de-trending of the

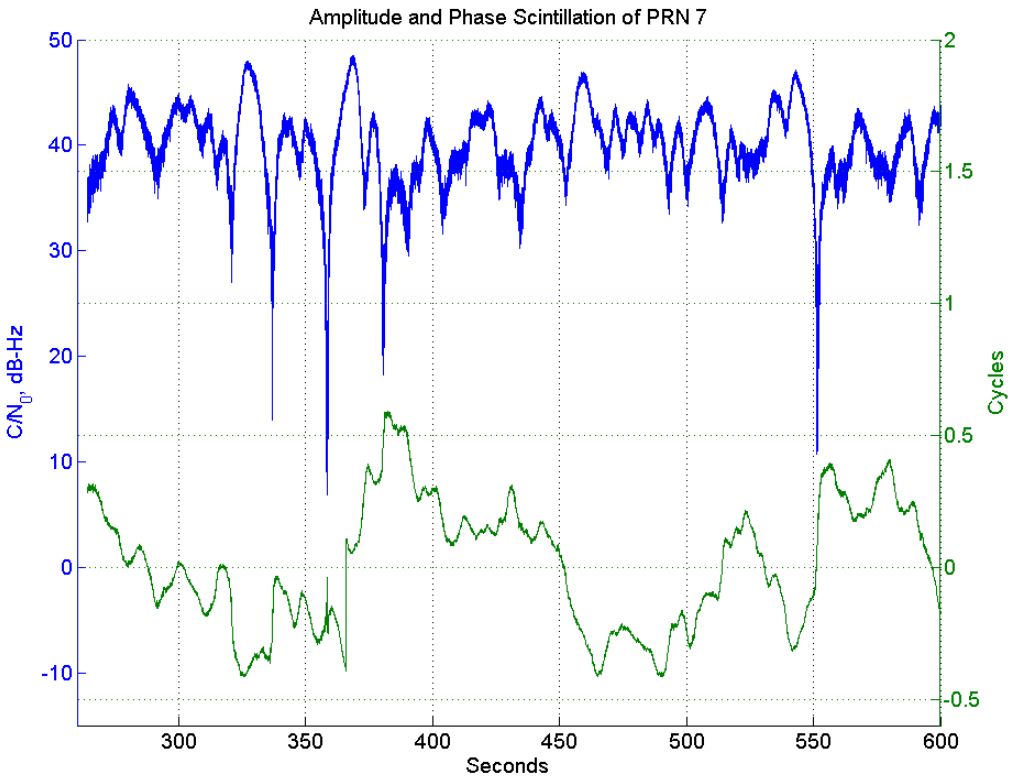

Figure 2.4: Amplitude and phase scintillation of a GPS signal.

phase measurement. The signal amplitude is shown in blue. This plot shows four easily recognizable "canonical fades" [18] (the abrupt half-cycle or nearly half-cycle phase shifts coincident with deep amplitude fades at approximately 320, 340, 380, and 500 seconds) as well as one severe amplitude fade that is not coincident with an abrupt half-cycle phase shift (at approximately 360 seconds).

2.4.2 An advanced triggering mechanism for determining the onset of scintillation

Due to the high data rates involved when logging data for scintillation study, it is desirable to have a reliable indicator for when signals are experiencing scintillation to avoid recording large amounts of uninteresting data. To put this in perspective, suppose a single "scintillation record" (e.g., amplitude, phase, and time stamp) takes 24 bytes to store. Recording these data for 24 individuals channels at 100 Hz requires about 5 GB per day. Processing this amount of data is prohibitive and even simply storing it quickly becomes onerous. Historically, receivers have used the aforementioned S_4 or σ_ϕ as triggering mechanisms: when one of these parameters exceeds some preset threshold, high rate data logging is begun. However, from a modeling standpoint it is desirable to have a single parameter that triggers the logging rather than some combination of two parameters. Further, σ_ϕ has been shown to be an unreliable indicator of scintillation intensity [19, 20]. These requirements led to the development of a spectrum-based triggering mechanism; this accounts for both amplitude and phase fluctuations, and a single triggering statistic can be computed by considering the ratio of power in a particular band to the total amount of power measured. This metric has been termed the "scintillation power ratio," or SPR. It should be noted that as this statistic includes the 100 Hz amplitude and phase data, the bandwidth is determined by the pre-detection bandwidth rather than the PLL bandwidth.

To compute the scintillation power ratio, the following steps are taken, using a 100 second window of data:

1. Remove local clock effects from carrier phase measurements and detrend, as in section 2.4.1, above.
2. Rotate the vector defined by the in-phase and quadrature accumulations by the phase from step 1. This results in a complex time history of the signal, with variations due only to ionospheric effects and noise terms.
3. Apply a windowing function (CASES uses a Hann window).
4. Take the FFT of the result.
5. Compute the ratio of the power in a particular frequency band to the total power in the accumulation bandwidth.

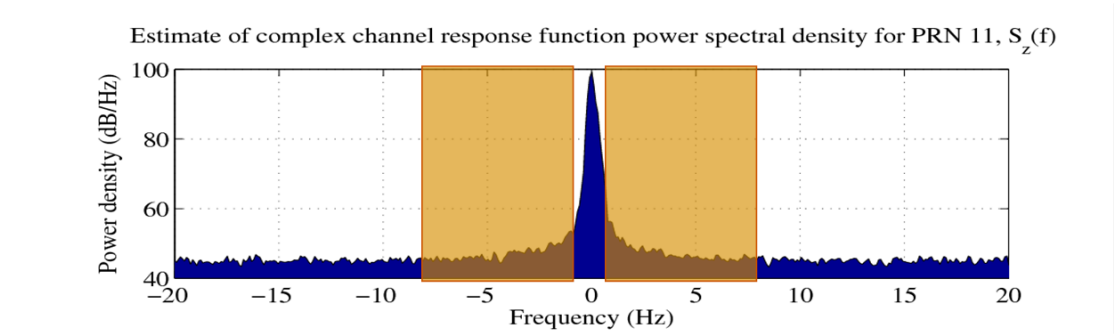

Figure 2.5: Power spectrum of the complex channel response functions of a scintillating signal. Frequency bands used for the scintillation power ratio are shown in orange.

The frequency band used for triggering is set to $\pm(0.2 - 8)$ Hz, though the user can change this. A power ratio in excess of -20 dB is taken to indicate a signal is experiencing scintillation. Fig. 2.5 shows the complex channel response function power spectral density (i.e., the result produced after step 4 of the above algorithm) for a GPS signal that was experiencing scintillation. There is a large DC component to this signal due to the direct component of the channel response function [20]. The frequency bands used in the SPR calculation are highlighted in this figure. These bands as well as the power ratio threshold were

chosen by examining a large amount of actual equatorial scintillation data, as well as existing literature that has performed similar examinations [12, 16, 18]. Preliminary results from using CASES have shown that these frequency bands are also appropriate for studying high-latitude scintillation.

An elevation mask is used to exclude satellites below a particular elevation from the calculation in order to minimize contributions from multipath errors. The frequency band used in the SPR calculation can be set by the user to any value using a configuration file. Similarly, the user can select a different window length, triggering threshold, elevation mask angle, and frequency resolution for the FFT.

As promised, the issue of reference channel selection (as described in Sec. 2.4.1) will now be revisited. To locate a channel that is free of ionospheric effects, the SPR is calculated using every possible pair of channels that are tracking the same signal type (e.g, GPS L1 C/A) with one of the channels acting as reference. The pair of channels that produce the lowest SPR (and has an SPR below some much more stringent threshold than the threshold used for triggering) are both declared reference channels. The SPR for this pair of channels is re-checked each time SPR is calculated to make sure it remains below the reference power threshold. If it exceeds that threshold, it is assumed that one or both of the reference channels are scintillating, and a new set of reference channels is searched for. If all channels are scintillating or if all channels except one are scintillating, then this technique for removal of clock effects will not work.

2.4.3 Data Buffering

As discussed in subsection 2.4.2 above, triggering of high rate data logging is used to effectively filter out "uninteresting" data, and minimize storage and processing requirements. This triggering method operates on batches of data 100 seconds in length for CASES, but window lengths of 60 seconds are common [17]. The result of this is that by the time high-rate logging is triggered, some time has elapsed since the onset of scintillation, and in the worst case an entire window period has passed. As this receiver was designed to advance the study of scintillation (among other goals), it seems prudent to provide the greatest amount of data from these events as is possible. Further, studying the onset of these events may prove critical to understanding the underlying atmospheric dynamics. With that in mind, a buffering scheme was implemented whereby data from all satellites is stored in a circular buffer (i.e., first in, first out) for 220 seconds. If a scintillation event is detected, the receiver outputs the data in the buffer for the scintillating signal. This is illustrated in Fig. 2.6. The data in Fig. 2.6 are actual (mild) amplitude scintillation gathered by the author during a campaign in March, 2011 in Lima, Peru. Suppose the triggering mechanism used the window indicated by the highlighted region for detection. In most receivers, if the indicated amplitude fade caused the detection statistic to trigger high-rate logging, it would not begin until the end of the window (as data from the entire window are used in calculating the statistic). In so doing, potentially valuable data are thrown away. By buffering data, CASES is able to log from the beginning of the plot, two minutes prior to the event that indicated scintillation was occurring.

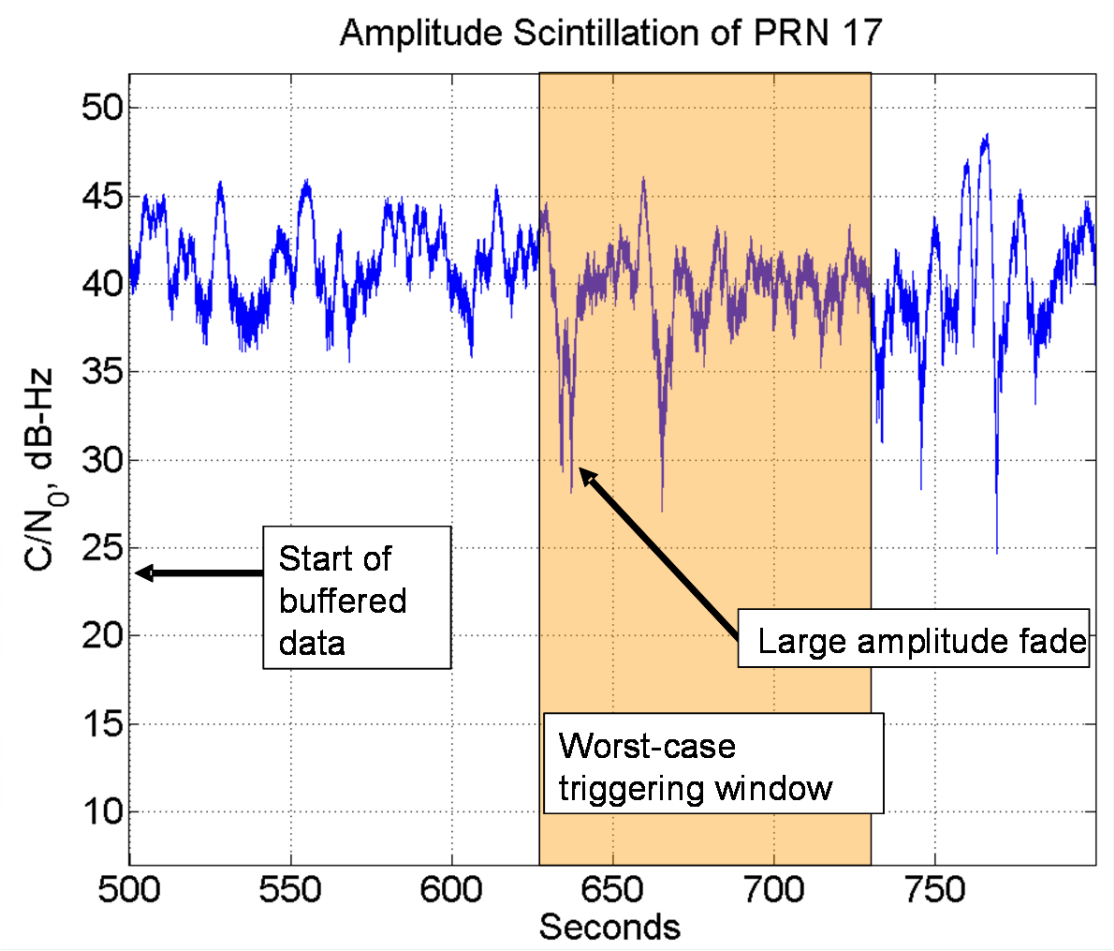

Figure 2.6: Illustration of the benefits of data buffering.

2.4.4 Data Bit Prediction

Scintillation-induced phase variations are particularly troublesome for the carrier tracking loops of GPS receivers, and present as a variety of phenomena including cycle-slipping and frequency unlock [13]. For a receiver designed to study scintillation effects, it behooves the designer to make the receiver as resilient to these effects as possible. GPS receivers generally operate with Costas-type PLL discriminators due to the modulation of the signal by the unknown 50 Hz data bit stream. This induces a loss of loop SNR known as squaring loss [21]. Carrier tracking performance can be improved with judicious choices for the

pre-detection interval, the loop bandwidth, and the loop discriminator [20]. CASES employs a 3rd order PLL with a decision-directed arctangent discriminator, a 7.5 Hz loop bandwidth, and a 10 millisecond pre-detection interval.

If the data bits are known *a priori*, a full-cycle (i.e., non-squaring) type PLL can be used, further improving tracking. This is particularly effective when in the presence of scintillation due to the aforementioned canonical fades that occur during scintillation, which manifest as half-cycle phase jumps. If the data bits are known, these phase jumps can be rightly measured as scintillation-induced variations rather than part of the signal. In the case of GPS, the 12.5 minute navigation message conveyed by the data bits changes quite infrequently (on even-numbered hours when the ephemeris data are updated or roughly daily in the case of almanac data). CASES records a library of observed data bits when the carrier-to-noise ratio is above a preset threshold, then uses these recorded data bits in the PLL if the carrier-to-noise ratio drops below that threshold (a possible indicator of scintillation). This data bit library also recomputes the time of week and parity data as required (as these are continually changing in a known manner), and monitors for possible ephemeris or almanac data updates. Results from testing the efficacy of the data-bit prediction algorithm are presented in Section 2.5. Note that there are small windows of time when the data bit library is unavailable, namely after an ephemeris or almanac data update, though the library makes it known that the bits are unavailable until the new data are recorded.

The L2 civil long signal is used when tracking on the GPS L2 frequency as this signal has no data bit modulation. Therefore, it is more robust to scintillation for the same reasons as given above.

2.5 Receiver Performance Analysis

The receiver has been run using both real and simulated[1] data in an effort both to confirm the operational advantages provided by the novel algorithms described herein, and to get a measure of the precision with which it can produce the standard observables such as phase and pseudorange.

2.5.1 Measurement Precision

The precision with which pseudorange can be measured is of particular importance as this impacts the accuracy of TEC estimates made using those measurements. The errors in carrier phase measurements are typically two orders of magnitude smaller than those for pseudorange [21].

To estimate the precision with which CASES can measure pseudorange, two receivers were connected to the same antenna, and TEC was computed using both pseudorange and carrier phase (for an entire satellite pass, about 5 hours). Fig. 2.7 illustrates this, with the pseudorange-derived value shown in blue, and carrier-phase-derived value shown in red.

The results for the two receivers are shown in the bottom two panels, while the difference of the results is shown in the top panel. The errors in the top panel are due only to receiver thermal noise and RF front-end differential filter delay effects. Local oscillator effects are removed by creating the differences shown in the bottom two panels as these errors are common to the measurements on L1 and L2. Multipath effects are eliminated in the inter-receiver difference as the

[1]Simulated RF data was generated by a Spirent GPS signal simulator, and contained signal scintillation as described by the model in Ref. [22].

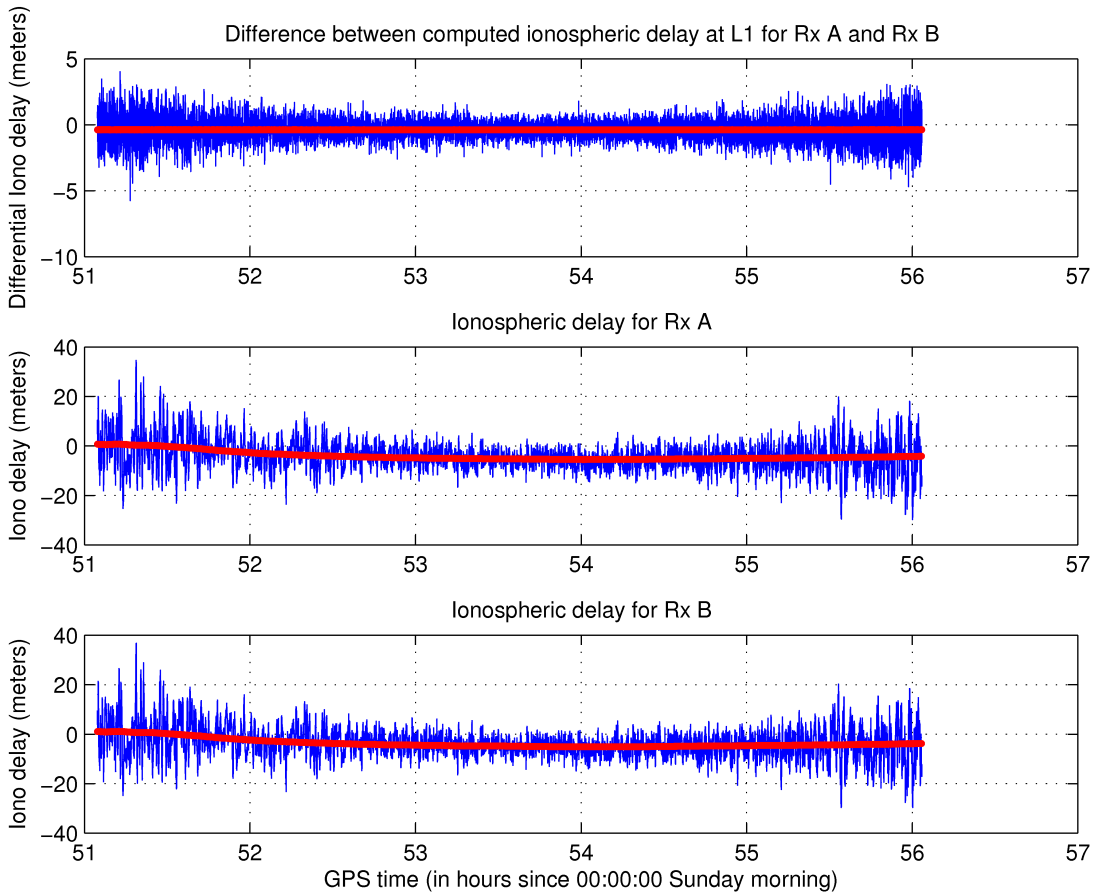

Figure 2.7: Single-receiver dual frequency ionospheric delay at L1 (bottom two panels) and inter-receiver ionospheric delay difference (top panel). Code-derived values are in blue and carrier-phase-derived values are in red. The receivers used a common antenna, and the carrier-phase-derived data have had a bias relative to the code-derived data removed.

two receivers shared an antenna. The increased noise at either end of the plot relative to the middle is due to lower C/N_0 and thus greater measurement noise. The minimum RMS noise in the top panel is 0.5 meters, which implies an RMS error for a single receiver of about 0.35 meters. This is the fundamental precision with which CASES can measure pseudorange on L1. Since the L2 civilian signal is weaker than the L1 C/A signal, and the receiver tracks only the civil long code on the L2 frequency, the L2 pseudorange measurements will be slightly worse than this.

Multipath errors can contribute as much as 5 meters (RMS) to pseudorange measurements [23]. The large multipath component is clearly shown in the bottom two panels as greatly increased noise as compared to the top panel (note the differing vertical scales). As CASES uses a narrow-bandwidth front-end, many advanced multipath mitigation techniques are unsuitable. One approach that is feasible is to tune the delay lock loop early-minus-late correlator spacing, which can result in better multipath rejection at the cost of tracking precision [24]. After testing, a value of 0.6 chips has been determined as optimal for the current receiver, and a (carrier aided) delay lock loop bandwidth of 0.1 Hz is used. The precision of CASES in the presence of multipath after tuning these parameters is shown in Fig. 2.8.

Again, pseudorange is shown in blue, carrier phase in red. In this plot, the two receivers (bottom two panels), are connected to different antennas. In this test, Rx A was in a good multipath environment, while Rx B was in a poor multipath environment (note the differing vertical scales). The RMS pseudorange error here for a single receiver (when the satellite was at high elevation) is 2.7 meters, but it should be noted that this is merely illustrating a typical value for pseudorange errors in the presence of multipath; the particular antenna used and the multipath environment are all significant factors here and any particular case could differ significantly.

2.5.2 Scintillation Robustness

Testing of the data bit prediction algorithm has shown that CASES is highly resistant to half-cycle phase jumps while experiencing ionospheric scintillation.

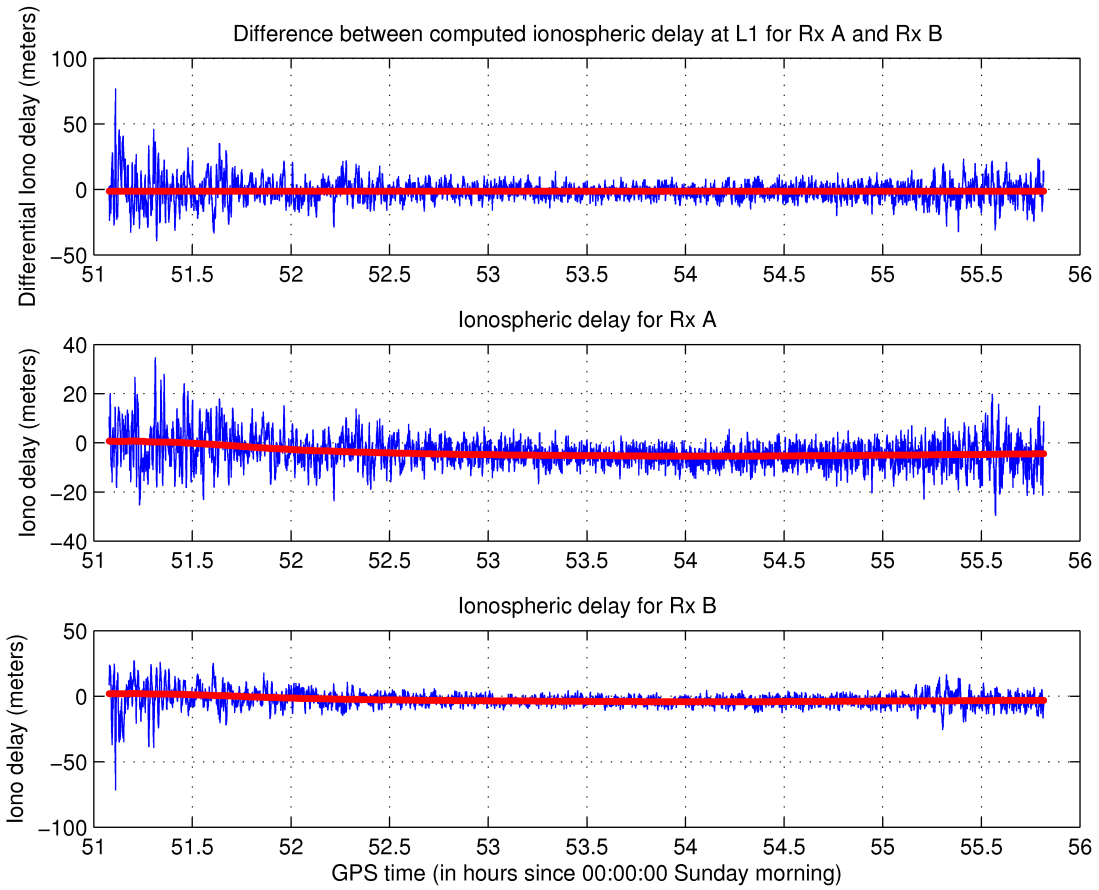

Figure 2.8: Single-receiver dual frequency ionospheric delay at L1 (bottom two panels) and inter-receiver ionospheric delay difference (top panel). Code-derived values are in blue and carrier-phase-derived values are in red. The receivers used independent antennas.

To conduct this test a scintillation scenario was generated using the Cornell Scintillation Model [22] on a Spirent GSS7700 GPS Signal Simulator. The simulation parameters were: nominal $C/N_0 = 43$ dB-Hz, $S_4 = 0.8$, $t_0 = 0.8$ s. The resultant signal was then tracked using CASES, and the measured phase history was subtracted from the true phase history recorded by the signal simulator. The CASES tracking was done in a post-processing mode after recording the data from the Spirent to ensure that exactly the same data were used for the comparison. The results of this test are shown in Figs. 2.9 and 2.10

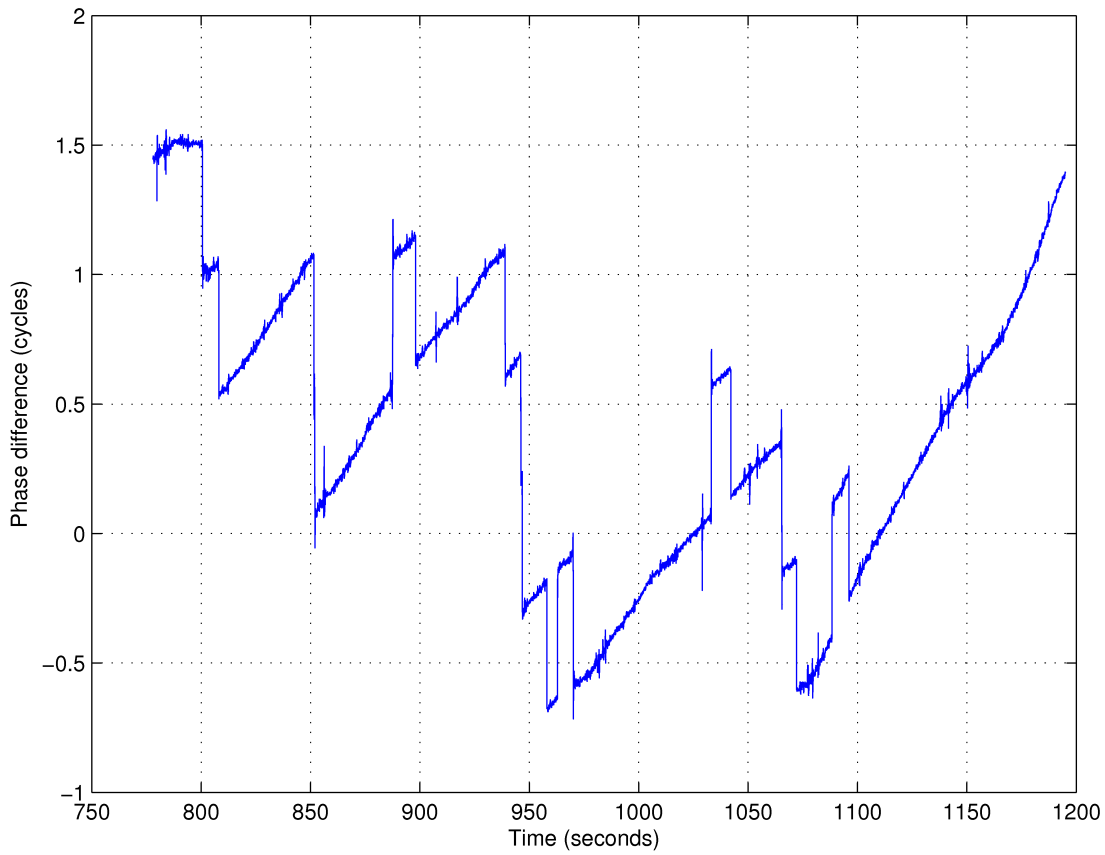

Figure 2.9: Phase error without use of data bit prediction.

The diagonal trend of the line sections is due to clock rate differences between the simulator and the receiver, and it should be noted that the vertical scales of the graphs differ in order to show as much detail as possible on each. Fig. 2.9 shows 16 half- or full-cycle slips. Fig. 2.10 shows only a single full cycle slip over the same period, thus it performs much better.

It should also be noted that while using the data bit prediction algorithm, only full cycle slips occur rather than half cycle slips, which are generally easier to remove in post-processing.

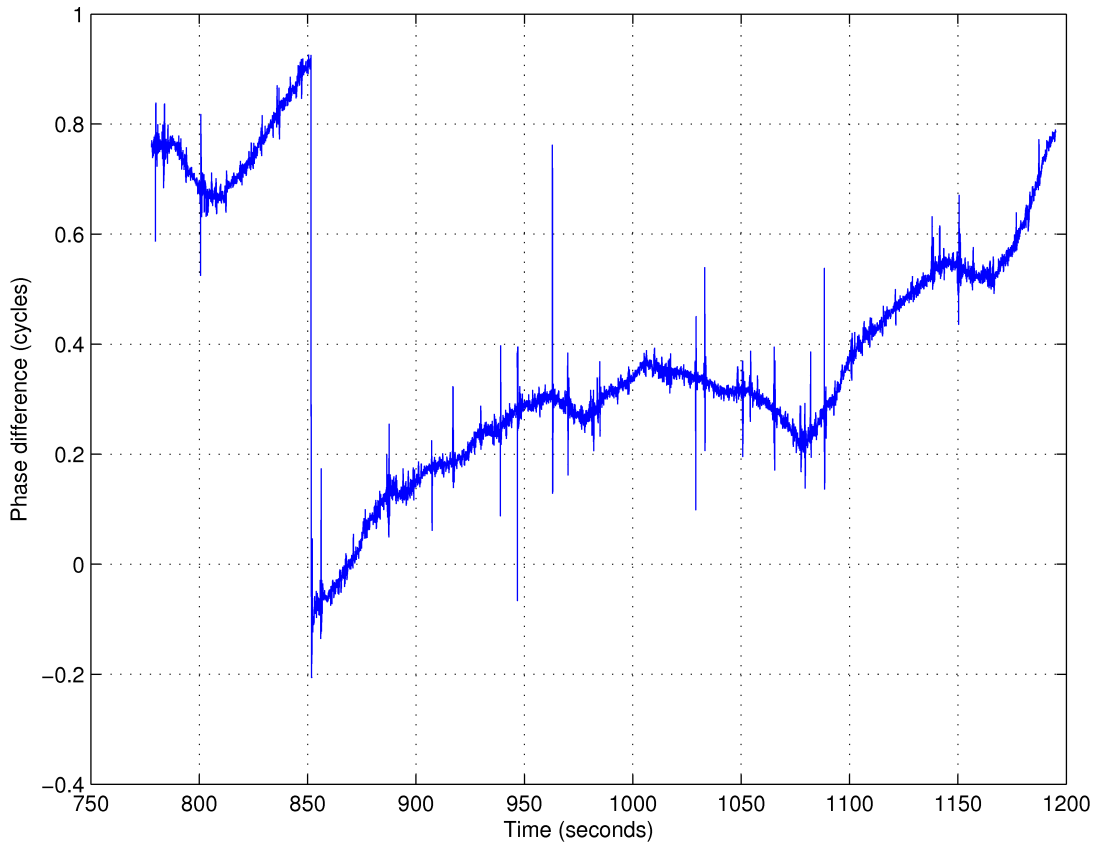

Figure 2.10: Phase error with data bit prediction.

2.5.3 Comparison With a Commercial Scintillation Monitor

CASES receivers were validated during a field campaign at the Jicamarca Radio Observatory near Lima, Peru in March, 2011. Six receivers were deployed in a small-baseline (1 km) array with the intent of observing scintillation and validating the ability of the receiver to operate while experiencing severe scintillation. Some observations recorded during this campaign have already been shown in Figs. 2.3, 2.4, and 2.6. One additional scintillation event is shown in Figs. 2.11 and 2.12. In Fig. 2.11, amplitude data from both CASES and a commercial scintillation monitor are shown in the top panel, with CASES in blue and the commercial receiver in red. Additionally, the bottom panel shows reported L1 C/A lock time for the commercial receiver. This plot illustrates

that the commercial receiver lost lock several times during the severe amplitude fades while CASES retained signal lock. S_4 during this period exceeded 0.9. A zoomed-in look at this plot between $700 - 770$ seconds is shown in Fig. 2.12. These plots show that CASES is capable of tracking through even quite vigorous scintillation.

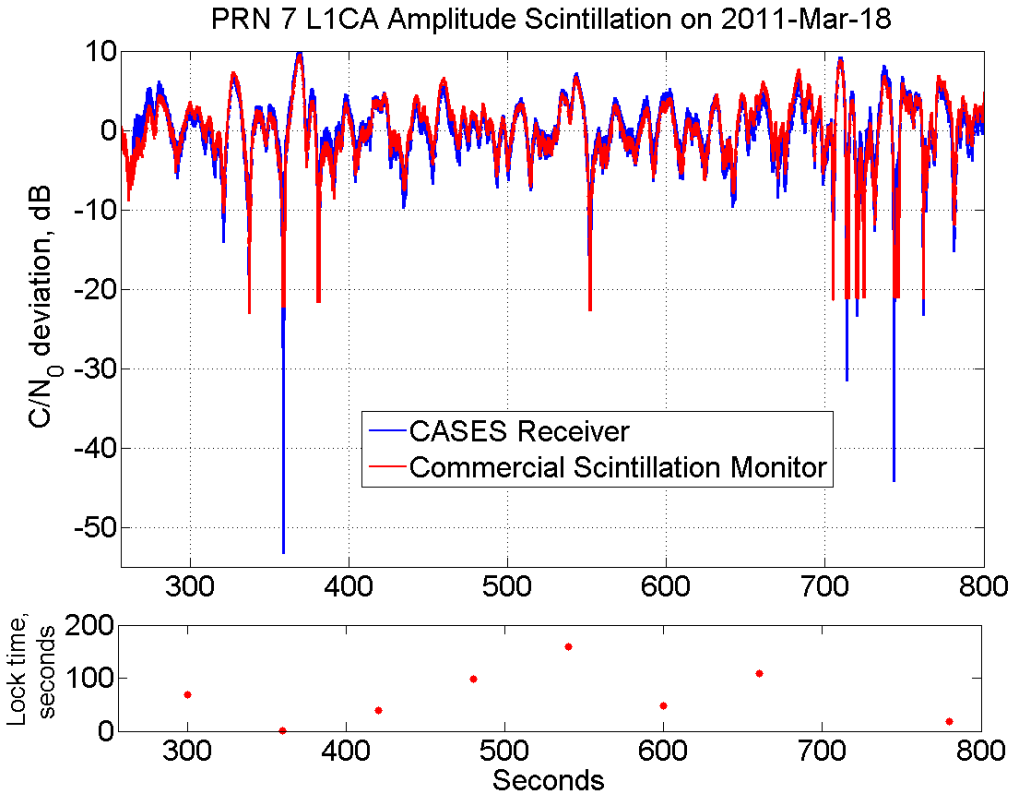

Figure 2.11: Amplitude scintillation observed by CASES and a commercial scintillation monitor.

2.6 Conclusions

A software-defined dual-frequency GPS receiver has been designed for use as a space weather monitoring instrument. This software has been embedded in a flexible and capable hardware platform that allows remote monitoring, data

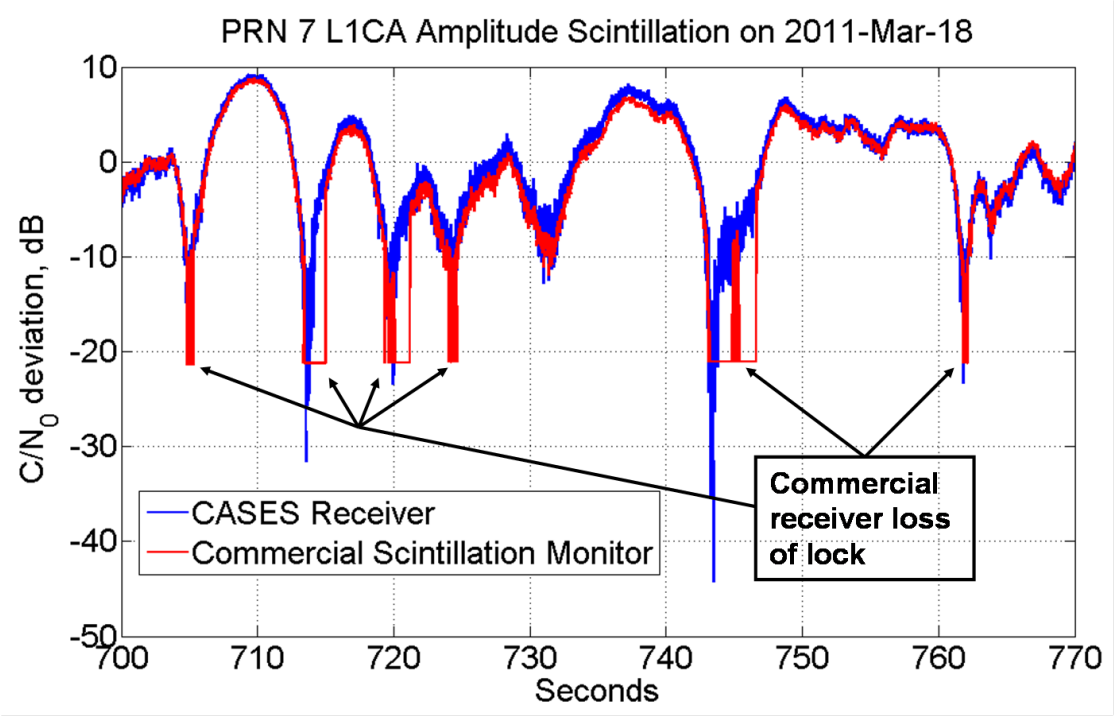

Figure 2.12: Detail of amplitude scintillation observed by CASES and a commercial scintillation monitor.

logging, and reconfiguration. This receiver implements several novel software processing techniques that allow it to excel at monitoring space weather due to an advanced triggering technique, special data buffering, removal of local clock effects, and a data bit prediction algorithm that makes it particularly robust to ionospheric scintillation. This platform has been tested both in the field and the laboratory and shown to have marked advantages versus receivers lacking these features.

CHAPTER 3

GPS SATELLITE ANOMALIES

B. W. O'Hanlon, "Carrier-Phase Anomalies Detected on SVN-48", *GPS World* via http://gpsworld.com/carrier-phase-anomalies-detected-on-svn-48/, May, 2010.

3.1 Introduction

Anomalous behavior of the L1 C/A code carrier phase has been detected on PRN 07/SVN 48. The anomalies are sudden step-like changes of phase by about 10deg/5mm. These steps are followed by negative steps of the same magnitude that restore the original phase time history. These anomalous square pulses have been observed with durations as short as 0.1 sec and as long as 600 seconds. They can occur about once a minute or be absent for hours.

This behavior was detected when testing a dual-frequency software receiver that processes the GPS civilian signals on L1 and L2. One use of this receiver is for making measurements of total electron content, or TEC. TEC, in so-called TEC units (1 TEC unit = 10^{16} electrons / m^2) is computed as

$$TEC = (\phi_{L1}\lambda_{L1} - \phi_{L2}\lambda_{L2}) \frac{f_{L1}^2 f_{L2}^2}{40.3\,(f_{L1}^2 - f_{L2}^2)} - b_{TEC} \tag{3.1}$$

where ϕ_{L1} and ϕ_{L2} are the beat carrier phase measurements of the signal received on the L1 and L2 frequencies, respectively, f_{L1} and f_{L2} are the L1 and L2 frequencies, respectively, the λ terms are the corresponding wavelengths, and

b_{TEC} is a bias term that occurs in the phase-based calculation and is unimportant here. The anomaly was first noted when calculating carrier-phase-based TEC, as the generally large but common-mode effects on carrier phase due to relative motion between the satellite and receiver are eliminated in the TEC calculation, allowing easy observation of this much more subtle effect.

3.2 Observations

A plot of the resulting TEC, after removal of its mean value, is shown in Fig. 3.1.

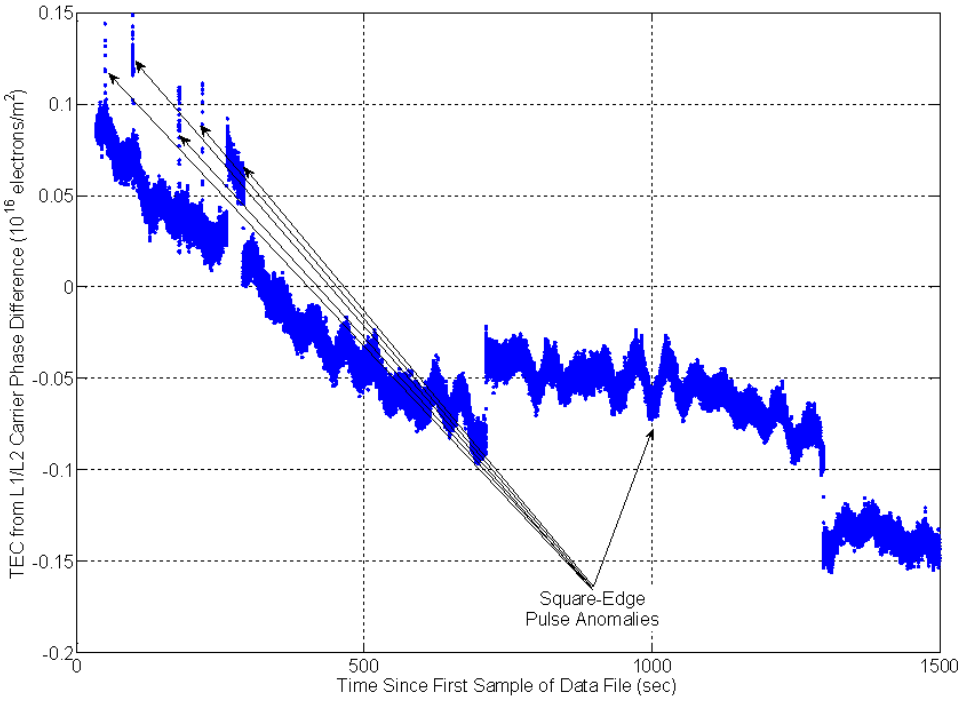

Figure 3.1: Square pulses on phase-based TEC due to L1 C/A carrier phase anomalies.

This figure shows 6 square-edged pulses that range in duration from 0.1 sec to 590 sec, with the first being a short one at t = 48 sec. The last pulse starts at t =

710 sec and ends at t = 1300 sec. In all cases, the anomaly consists of an apparent positive step change in TEC followed some time later by a negative step change of identical magnitude. Step magnitudes in the range 0.04 to 0.07 TEC Units have been observed.

Tests were performed in order to ascertain whether the anomalies were caused by the L1 signal, the L2 signal, or a combination of the two. Additional tests ruled out receiver malfunction as the cause of the anomalies.

Observation of detrended L1 and L2 carrier phase time histories quickly revealed that the anomalies occur on the L1 carrier phase. The detrended L1 C/A carrier phase shows square-edged pulses that corresponded to the times, magnitudes, and signs of the TEC anomalies, but the detrended L2C carrier phase plots show no such pulses. A typical detrended (negative) L1 C/A beat carrier phase anomaly is shown in Fig. 2, plotted in range-equivalent meters.

Extensive tests were performed in order to check whether the anomalies may have been caused by the receiver. The anomalies were initially discovered using a digital storage receiver of raw RF front-end samples followed by off-line software receiver processing. Such carrier phase anomalies could result from signal glitches in the RF front-end's mixing chain, from data recording anomalies in the RF front-end samples, or from errors in the software receiver code. The former two possibilities were ruled out by two means. One was to process signals from other satellites for the same RF samples. Mixing problems or data sample problems would cause similar anomalies on all GPS signals, but other GPS signals were found to be free of anomalies. Additional tests used simultaneous data collection by two digital storage receivers that were spaced 700 meters apart. The two devices also used different RF front-end hardware. Both

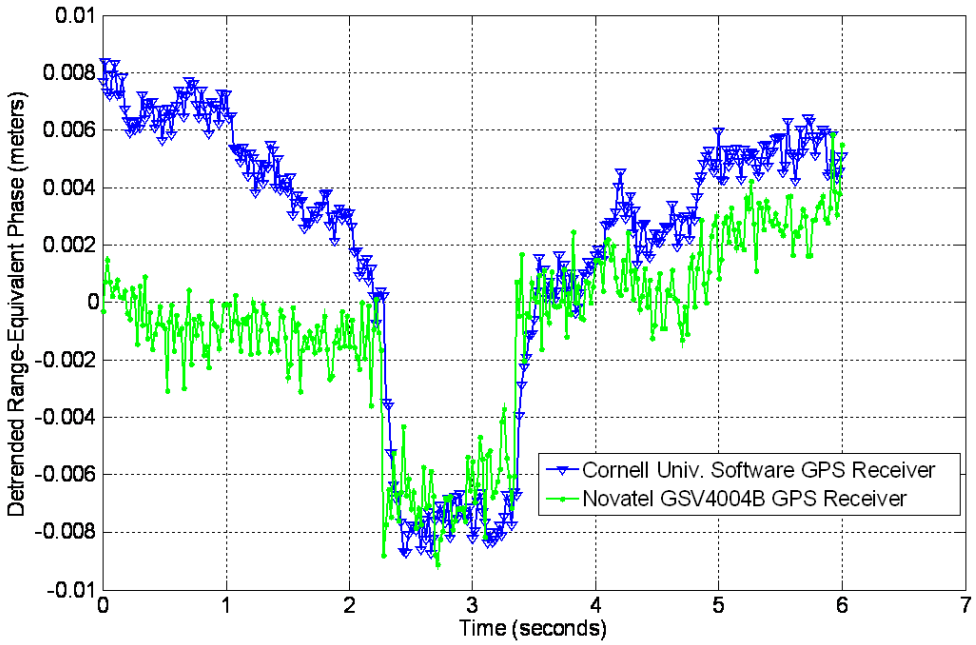

Figure 3.2: A typical anomaly as observed in the (negative) range-equivalent L1 C/A carrier phase .

receivers showed identical anomalies at identical times.

Software receiver code errors were ruled out by employing two independent sets of receiver processing code, one developed in MATLAB and the other in C. These two pieces of software were developed independently by different individuals and were run independently by their developers. They both showed identical anomalies. A final check used a completely different receiver, the commercially available Novatel GSV4004B. Figure 2 plots its detrended L1 C/A carrier phase is along with that of the C-based Cornell software receiver. They both show the same anomaly. Thus, they appear to be caused by the SVN 48 transmitter.

All of these observations have been made from roof-mounted antennas in Ithaca, N.Y. The anomalies were first observed on March 24, 2010, and were

observed again on April 1, 5, 7, and 29th. They have been observed as late as May 13. There was one period of several hours on May 11 when no anomalies occurred. The following additional Block IIR-M satellites have also been monitored briefly, but without finding any similar anomalies to date: SVN 58/PRN 12, SVN 55/PRN 15, SVN 57/PRN 29, SVN 49/PRN 01, and SVN 50/PRN 05.

3.3 Conclusion

These anomalies could be of consequence for some GNSS applications. For precise monitoring of differential TEC, the magnitude of this anomaly is the same order as the signals of interest. Precise Point Positioning (PPP) systems seek to achieve CDGPS accuracy without direct double differencing. The lack of double differencing would allow any L1 C/A carrier phase anomaly to directly affect the PPP solution.

CHAPTER 4

REAL-TIME GPS SPOOFING DETECTION VIA CORRELATION OF ENCRYPTED SIGNALS

B.W O'Hanlon, M.L. Psiaki, J.A. Bhatti, D.P. Shepard, and T.E. Humphreys, "Real-Time GPS Spoofing Detection via Correlation of Encrypted Signals," *NAVIGATION, Journal of the Institute of Navigation*, vol. 60, no. 4, pp267-278, 2013.

4.1 Abstract

A method for detecting the spoofing of civilian GPS signals has been implemented and successfully tested in a real-time system. GPS signal spoofing is an attack method whereby a third party transmits a signal that appears authentic but induces the receiver under attack to compute an erroneous navigation solution, time, or both. The detection system described herein provides a defense against such attacks. It makes use of correlations between the unknown encrypted GPS L1 P(Y) code signals from two narrow-band civilian receivers to verify the presence or absence of spoofing. One of these receivers is assumed to be at a secure location that is not subject to spoofing. The other receiver is the potential spoofing victim for which the present developments constitute a defense. Successful detection results are presented using a reference receiver in Ithaca, New York, a victim receiver in Austin, Texas, and a spoofer in Austin, Texas.

4.2 Introduction

GPS spoofing is a method of attacking a Global Positioning System receiver with the goal of having the targeted receiver compute an erroneous navigation solution, an incorrect time, or both [3, 4, 5]. Spoofing of the unencrypted civilian L1 C/A signal has been demonstrated both in the laboratory [4, 5] and in the field [25]. Although there have, as yet, been no confirmed GPS spoofing attacks observed "in the wild," the vulnerability of GPS to spoofing attacks has long been a concern [3, 2, 26], and an Iranian engineer claimed to have used GPS spoofing to capture a highly classified U.S. drone in December, 2011 [27]. Proposed methods for detecting spoofing attacks include examining changes to certain signal characteristics [28], incorporation of external hardware such as an inertial measurement unit [29], use of multiple [30] or moving receiver antennas [31], or cryptographic techniques [32, 33, 34, 8, 35, 36].

This work is an extension of the work presented in Ref. [33] and presents an implementation of the cryptographic technique of Refs. [32, 33, 34, 8] that operates in real-time. A major contribution of this work is that it is the first real-time implementation of a cryptographic civilian GPS spoofing detection technique. Tests of this system against a sophisticated spoofer constitute another significant contribution.

This spoofing detection technique makes use of the encrypted P(Y) code and assumes that it cannot be spoofed. Given that the U.S. military has implemented and guards this encryption, this is a reasonable assumption, with two caveats: this technique does not detect "meaconing" [36] attacks, or attacks that attempt to estimate and replay the P(Y) encryption code. The former attack is fairly

limited in scope, and there are other defenses [36] against the latter attack, thus this defense still has significant value.

Given the assumed security of the P(Y) code, an attacker spoofing the civilian C/A code will be unable to provide spoofed P(Y) code that has the correct encryption chips. The two receivers in this spoofing detection system know the expected relationship between the C/A code and the P(Y) code. They both track the C/A code and isolate the component of the signal that should be a noisy version of the P(Y) code. A communication link is used to send one of these signal components to the other receiver, which then correlates the two versions to produce a spoofing detection statistic. In this implementation, the raw unprocessed samples from one receiver front-end are transmitted to the other receiver (which then does all of the aforementioned processing), as this is more efficient when more than one satellite is being tracked.

A large value of the spoofing detection statistic indicates identical true P(Y) code in both receivers, which indicates that the potential victim receiver (the "defended receiver") is not being spoofed. A near-zero value of this statistic, on the other hand, indicates the absence of P(Y) code in one or both receivers, presumably just in the victim receiver. This low value indicates a spoofing attack. Of course, P(Y) code could be missing from the other receiver (the "reference receiver"), but this system assumes the reference receiver has been made impervious to spoofing. The choice of a detection threshold against which this statistic is compared is addressed with a hypothesis testing analysis. The hypothesis testing analysis used here is an adaptation of the one developed in Ref. [8]. Adaptation has been necessitated by simplifications in the test statistic calculation which facilitate real-time operation. The simplified statistic calcula-

tion and hypothesis test analysis constitute another contribution of this paper.

In the course of testing this algorithm, it has been necessary to perform an experimental calibration of the difference in transmitted power between the L1 C/A and P(Y) code signals. A nominal value is suggested in the GPS Interface Specification [37], but experimental results have shown that it differs slightly from the published value, and the value differs somewhat between satellites. These variations have been previously observed [38, 39], but the current satellite constellation differs significantly from its state at the time of that work. Another contribution of this paper is a table with the experimentally-derived transmission power differences for each satellite.

The remainder of the paper is organized as follows. The first section discusses the experimental set-up and the hardware used for receiving and processing the data. The second section reviews the signal model and necessary pre-processing and introduces two new loss factors that are required for correct estimation of the P(Y) signal power. Next, the algorithm for computing the spoofing test statistic and other quantities required for the hypothesis test is described. Results from testing this system under a spoofing attack are then presented. Also included in the results section is a table with the measured per-satellite difference in L1 P(Y) code vs. C/A code signal power. The paper concludes with a discussion and summary of its important points.

4.3 Architecture

The spoofing detection system consists of two narrow-band radio front-ends and data capture systems, a full GPS software radio system with additional

spoofing detection software running on a personal computer, and a method for transporting the data from the two front-ends to the software radio computer. Although the radio front-end/data capture systems are not stand-alone GPS receivers, they will nonetheless be referred to as "receivers" in the interest of brevity.

Both receivers perform mixing, filtering, and sampling of the radio-frequency signal, producing 2-bit samples at a sample rate of 5.7 MHz. The intermediate frequency filters in the two front-ends each have a nominal 3-dB bandwidth of 2.5 MHz. This narrow bandwidth attenuates the wide-band P(Y) signal by 5.5 dB and distorts it. The exact filter response for each receiver has been characterised using off-line system identification techniques [40]. Knowledge of this response is important for the correct design of a spoofing detection hypothesis test that properly accounts for the P(Y) signal attenuation and distortion.

The data from each receiver are either directly recorded to a hard drive, or streamed over a network connection and then recorded to a hard drive, depending on the particular equipment set-up. The software receiver and spoofing detection calculations operate on the data that is stored to disk. It can operate in a real-time mode or an after-the-fact mode. It would be possible to replace the disk storage with a buffer in RAM.

To reduce the possibility of a man-in-the-middle type attack where the data transmitted over the network are intercepted by a third party and replaced with other (possibly spoofed) data, a secure shell (SSH) tunnel was used for all network communication. This encrypts the data using triple DES encryption and is believed to be secure [41].

Software that carries out the spoofing detection calculations has been added to an existing GPS software receiver [42]. This software processes the samples from both front-ends in parallel. It tracks the civilian L1 C/A code signals, performs P(Y) code cross-correlation, and produces a metric that indicates the likelihood that the defended receiver is being spoofed. This software receiver could be physically located anywhere; for this system it was co-located with the reference receiver for convenience and the raw unprocessed samples from the defended receiver were transmitted to it over the Internet. By transmitting the unprocessed samples rather than the processed signal component for each individual signal, the required communications bandwidth is reduced. The receiver architecture used for this work is illustrated in Fig. 4.1. Except for the SSH tunnel and the fact that the software receiver simultaneously processes data from two radio front-ends, everything to the left of the dashed vertical line represents standard software radio hardware and code. Everything to the right of that line represents new calculations needed for spoofing detection.

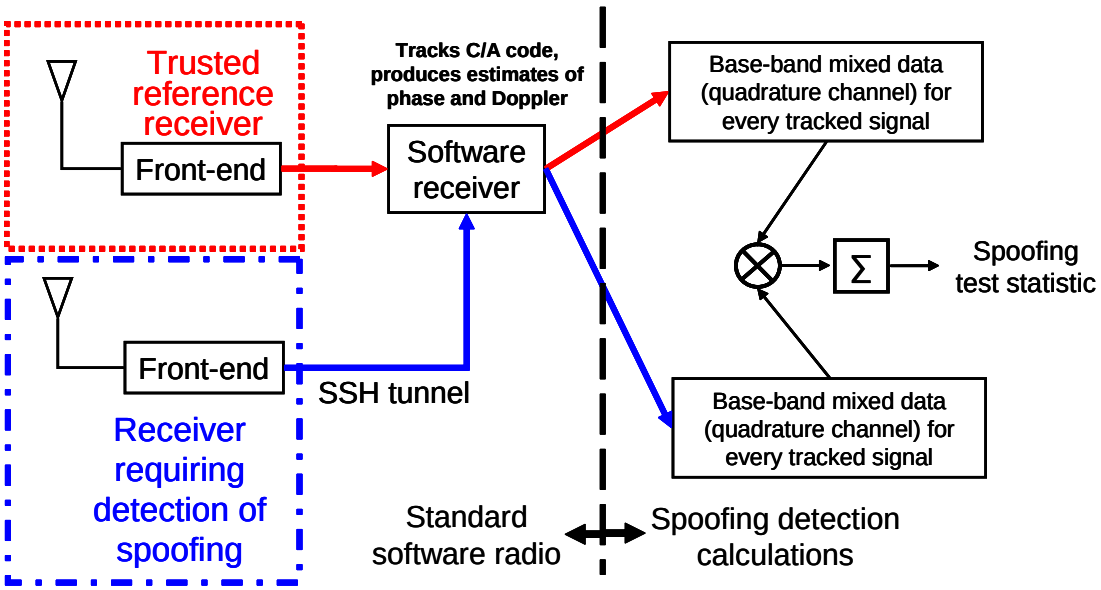

Figure 4.1: An example receiver architecture.

All processing for this work was done on a personal computer with a quad-

core Intel i7 930 CPU. This implementation processes data at approximately 3 times faster than real-time (i.e., it processes 30 seconds of data in 10 seconds); up to 30 satellites common to both receivers can be tracked simultaneously in real-time. Standard hemispherical patch antennas were used at both the reference and defended receivers.

4.4 Signal Model and Pre-Processing

The pre-processing and signal model used here are developed in Ref. [8]. For convenience, the relevant subset of equations from Ref. [8] are reproduced in this section.

4.4.1 Signal Model

The models of the signals at the outputs of the RF fronts ends of the two receivers take the form:

$$\begin{aligned} y_{Ai} =& A_{cA} C_f(t_{Ai}) D(t_{Ai}) \cos[\omega_{IF} t_{Ai} + \phi_A(t_{Ai})] + \\ & A_{pA} P_{Yf}(t_{Ai}) D(t_{Ai}) \sin[\omega_{IF} t_{Ai} + \phi_A(t_{Ai})] + n_{Ai} \end{aligned} \tag{4.1a}$$

$$\begin{aligned} y_{Bi} =& A_{cB} C_f(t_{Bi}) D(t_{Bi}) \cos[\omega_{IF} t_{Bi} + \phi_B(t_{Bi})] + \\ & A_{pB} P_{Yf}(t_{Bi}) D(t_{Bi}) \sin[\omega_{IF} t_{Bi} + \phi_B(t_{Bi})] + n_{Bi} \end{aligned} \tag{4.1b}$$

where y_{Ai} is the sample output by the RF front-end of receiver A, the reference receiver, at receiver A clock time t_{Ai}, and y_{Bi} is the output of receiver B, the defended receiver, at receiver B clock time t_{Bi}.

$C_f(t)$ and $P_{Yf}(t)$ are functions representing the C/A and P(Y) codes, respectively, after attenuation and distortion by the RF front-end. The function $D(t)$ represents the 50 Hz navigation data bit modulation. A_{cA} and A_{cB} are the received C/A code amplitudes for receivers A and B, and the corresponding P(Y) code amplitudes are A_{pA} and A_{pB}.

The nominal intermediate frequency is ω_{IF}, and $\phi_A(t)$ and $\phi_B(t)$ are the beat carrier phase time histories of the signals at the two receivers. The time derivatives of $\phi_A(t)$ and $\phi_B(t)$ equal the receiver carrier Doppler shifts.

The remaining terms, n_{Ai} and n_{Bi}, are the receiver noise terms, which are assumed to be discrete-time Gaussian white-noise with statistics:

$$E(n_{Ai}) = 0, E(n_{Ai}^2) = \sigma_{RFA}^2, E(n_{Ai}n_{Aj}) = 0 \text{ for all } i \neq j \tag{4.2a}$$

$$E(n_{Bi}) = 0, E(n_{Bi}^2) = \sigma_{RFB}^2, E(n_{Bi}n_{Bj}) = 0 \text{ for all } i \neq j \tag{4.2b}$$

$$E(n_{Ai}n_{Bi}) = 0 \text{ for all } i, j \tag{4.2c}$$

4.4.2 Tracking the C/A Signal

To process the C/A code, the signal from the front end is mixed with carrier and code replicas and accumulated for some period. These accumulations are then used in discriminators in a Delay-Lock-Loop (DLL) and a Phase-Lock-Loop (PLL) for feedback-based tracking. The prompt in-phase and quadrature

accumulations for the k^{th} accumulation interval are:

$$I_k = \sum_{i=i_k}^{i_k+N-1} y_i C[(i\delta T - \tau_k)(1 + \hat{\omega}_{Dk}/\omega_{L1})] \times \cos(\omega_{IF} i\delta T + \hat{\phi}_k + \hat{\omega}_{Dk}(i\delta T - \tau_k)) \tag{4.3a}$$

$$Q_k = \sum_{i=i_k}^{i_k+N-1} y_i C[(i\delta T - \tau_k)(1 + \hat{\omega}_{Dk}/\omega_{L1})] \times \sin(\omega_{IF} i\delta T + \hat{\phi}_k + \hat{\omega}_{Dk}(i\delta T - \tau_k)) \tag{4.3b}$$

where δT is the nominal front-end sampling period, τ_k is the DLL-produced estimate of the start time of the first code period in the accumulation as measured by the receiver clock, and the sample index i_k is the first sample in that C/A code period (i.e., the first sample such that $i_k \delta T \geq \tau_k$). The number of samples in the accumulation is N_k, which can be approximated as N, a constant. The function $C[t]$ is the local replica of the PRN code without RF filter effects; it takes on the values ± 1. The PLL's carrier Doppler shift estimate for the k^{th} code period is $\hat{\omega}_{Dk}$, and $\hat{\phi}_k$ is the estimated beat carrier phase at the code period start time τ_k. The subscripts A and B have been omitted here as the processing is identical for both receivers.

The spoofing detection algorithm exploits the known phase-quadrature relationship of the encrypted P(Y) code relative to the C/A code, as per Eq. 4.1. It isolates the quadrature part of the signal with the help of the PLL and DLL outputs $\hat{\phi}$, $\hat{\omega}_{Dk}$, and τ_k. The C/A code accumulations in Eq. 4.3 are used to estimate the carrier-to-noise ratios required by the hypothesis test.

4.4.3 Received Signal Power

The received carrier-to-noise ratio of the P(Y) signals in the two receivers are important inputs to the spoofing detection calculations. These quantities can be inferred based on measured C/A code carrier-to-noise ratios coupled with knowledge of receiver properties and calibrated relationships between the transmitted C/A code power and P(Y) code power for the various GPS satellites.

The inputs to the C/A carrier-to-noise calculation are time histories of the I_k and Q_k prompt accumulations. These values can be used to estimate the amplitude of the [I_k;Q_k] vector and the variance of the Gaussian noise in each individual I_k and Q_k accumulation. These estimates are:

$$A_{IQ} = (\bar{z}^2 - \sigma_z^2)^{1/4} \tag{4.4a}$$

$$\sigma_{IQ}^2 = 0.5(\bar{z} - \sqrt{\bar{z}^2 - \sigma_z^2}) \tag{4.4b}$$

where $\bar{z}$ is the mean of the accumulation power and σ_z^2 is its variance. These are calculated from the raw accumulations as follows:

$$\bar{z} = \frac{1}{K}\sum_{k=1}^{K}(I_k^2 + Q_k^2) \cong E\{I_k^2 + Q_k^2\} \tag{4.5a}$$

$$\sigma_z^2 = \frac{1}{K}\sum_{k=1}^{K}(I_k^2 + Q_k^2)^2 - \bar{z}^2 \cong E\{[I_k^2 + Q_k^2]^2\} - \bar{z}^2 \tag{4.5b}$$

where K is the number of prompt accumulations included in these averages. For this work, $K = 1000$ has been used with one millisecond accumulations. These calculations can be used to estimate the effective variance of the noise in the raw RF samples:

$$\sigma_{RF}^2 = \frac{2}{N}\sigma_{IQ}^2 \tag{4.6}$$

This notation differs somewhat from Ref. [8] because here the number of samples in each accumulation, N, is constant, whereas in that work it was allowed to vary.

The C/A code carrier-to-noise ratio is computed using the quantities in Eq. 4.4 as:

$$(C/N_0)_c = \frac{A_{IQ}^2}{2\sigma_{IQ}^2 T_{accum}} \tag{4.7}$$

where $T_{accum} = \delta t N$ is the accumulation period.

The P(Y) code carrier-to-noise ratio can be computed from the C/A carrier-to-noise ratio, but several loss factors must be taken into account. These loss factors account for the effect of using un-filtered C/A code in Eq. 4.3 and the effects of the front-end filtering on the C/A signal. Let the combined effect of these two influences be denoted L_{fca}. The effect of the RF filter on the received P(Y) signal produces the loss L_{fpy}. An additional loss factor, L_{psv}, represents the difference in transmitted power between the P(Y) and C/A signals. This is nominally 3 dB, with P(Y) power lower than C/A power [37]. It has been discovered to vary between satellites and to differ somewhat from the nominal value. A contribution of this work is the experimental calibration of this previously unknown variation between satellites.

The P(Y) code carrier-to-noise ratio is:

$$(C/N_0)_{py} = L_{psv} L_{fpy} \left[\frac{10^{-0.04/10} (C/N_0)_c}{L_{fca}} \right] \tag{4.8}$$

The method of calculating L_{fpy}, L_{fca}, and the power of 10 are explained in Ref. [8]. The values for L_{psv} are given in a table in Sec. 4.6. Excluding the contribution of L_{psv}, the factor $L_{fpy} \frac{10^{-0.04/10}}{L_{fca}}$ has been computed to be -5.06 dB

and -4.92 dB for the reference and defended receivers, respectively.

4.5 Methodology

This section explains the real-time implementation of the spoofing detection technique presented in Refs. [32, 33, 34, 8]. Several of the modifications needed for practical real-time implementation are non-trivial.

4.5.1 Spoofing Detection Statistic Calculation

To detect if the defended receiver is being spoofed, a spoofing detection statistic as in Ref. [8] has been produced. To accomplish this, the part of the signal from each receiver that is in quadrature with the C/A code is isolated, and the two quadrature data streams from the two receivers are mixed together and accumulated. Doing this requires knowledge of the carrier Doppler shift $\hat{\omega}_D$, carrier phase $\hat{\phi}$, and code start time estimate τ for each signal from both receivers. It also requires the data from both receivers to have been aligned in time, i.e., a given pair of mixed quadrature samples from the two receivers must have the same time of transmission.

To accomplish time alignment, the software first performs a coarse synchronization by locating the beginning of the same C/A code period in the data from the two receivers for each signal. This is done by decoding the transmitted data bits from each receiver to calculate the GPS time of week which indicates the transmission time of a particular code period [37]. Each channel then processes however many C/A code periods worth of data are required such that at the

end of this process the reference and defended receiver channels are all processing the C/A code period corresponding to the same nominal transmit time (as this is a software receiver, "channel" in this context means the code that handles the signal from a particular satellite). In general this requires a small amount of waiting for the desired data to become available, depending on how closely synchronized the data gathering is at the two receivers, on the receiver-satellite geometry, and on network latency. The delay is generally a small fraction of a second, so the process can still be considered "real-time."

After this coarse synchronization, the reference and defended receiver channels each process a single C/A code period, followed by a P(Y) code cross-correlation for the same period. To perform this cross-correlation, the quadrature part of the signal from each channel must be isolated, and a set of samples from the reference receiver must be somehow matched to a set of samples from the defended receiver. This sample matching is, in effect, a fine time alignment between the two receivers that attempts to match samples as closely as possible by the time of transmission of the underlying waveform. As the receiver is tracking the phase of the L1 C/A signal with a PLL in the normal course of operation, the correct carrier phase $\hat{\phi}$ and frequency $\hat{\omega}_D$ required for quadrature mixing are known.

The code start time estimate τ is also known for each receiver, but these times do not directly indicate the optimal matching of samples between the two receivers for cross-correlation. To determine the correct sample matching between the two receivers it is necessary to consider the effects of relative motion between the satellites and those receivers as it affects their relative Doppler shifts. In general, the carrier Doppler shifts of the signals from each receiver

will differ. To produce a P(Y) code cross-correlation with maximum power, the nominal alignment of data between the two receivers is selected such that the midpoint of their C/A code periods are aligned. This causes the P(Y) codes to be aligned as well.

It should be noted that in the software receiver implementation many variables are stored as fixed-point for purposes of processing efficiency. In particular, the code start time estimate τ is stored as an integer sample index i_k plus a fractional sample index i_{k-frac}:

$$\tau = \delta T \left(i_k + \frac{i_{k-frac}}{S_T} \right) \tag{4.9}$$

where δT is the front-end sampling period. The variable i_{k-frac} is always non-negative (though it is stored as a signed quantity), and scaled by a factor S_T such that 1 sample has S_T subdivisions. In general, a given C/A code period has initial sample indices i_{kA} and i_{kB} for receivers A and B, respectively. It also has fractional sample counts $i_{kA-frac}$ and $i_{kB-frac}$. Also note that cross-correlation is done on a single C/A code period at a time (a "sub-accumulation"), and these sub-accumulations are summed over many code periods to produce the final result.

In the simplest case, $i_{kA-frac}$ and $i_{kB-frac}$ are equal, and the carrier Doppler shifts in the two receivers are identical. Sample index matching is simple in this case: the quadrature sample with index i_{kA} in receiver A is mixed with the quadrature sample with index i_{kB} from receiver B, sample $i_{kA} + 1$ in receiver A is mixed with sample $i_{kB} + 1$ in receiver B, and so on. The general case of non-equal i_{k-frac} values and Doppler shifts requires more consideration.

In correcting for the signal Doppler shift, an "effective code start time" is

calculated. This effective code start time indicates the sample time that is, considering the effect of Doppler shift, exactly one-half of a C/A code period away from the midpoint of the accumulation interval. This correction manifests as a correction to the fractional sample index i_{k-frac} as follows. Given the number of samples in the code-period accumulation, N, the nominal L1 frequency ω_{L1}, the signal Doppler shift frequency, $\hat{\omega}_D$, the sample period δT, and the nominal C/A code chipping rate F_{c-nom}, the correction to the fractional code start time estimate is:

$$\delta i_{k-frac} = round\left(\frac{N\delta T\left[F_{c-nom}\left(1+\frac{\hat{\omega}_D}{\omega_{L1}}\right)-F_{c-nom}\right]S_T}{2F_{c-nom}\delta T}\right) = round\left[\frac{N\hat{\omega}_D S_T}{2\omega_{L1}}\right] \tag{4.10}$$

For $N\delta T \approx 1$ millisecond and a stationary receiver, $|\delta i_{k-frac}| << S_T$. This correction is subtracted from the current fractional code start time to give an effective fractional code start time:

$$\hat{i}_{k-frac} = i_{k-frac} - \delta i_{k-frac} \tag{4.11}$$

Note that this new $\hat{i}_{k-frac}$ could be negative. If $\hat{i}_{k-frac} < 0$, $\hat{i}_{k-frac}$ is incremented by S_T and the effective integer sample index is $\hat{i}_k = i_k - 1$. If $\hat{i}_{k-frac} > S_T$, $\hat{i}_{k-frac}$ is decremented by S_T and $\hat{i}_k = i_k + 1$. Otherwise the effective integer sample index is $\hat{i}_k = i_k$. Equations 4.10 and 4.11 are applied to the data from receivers A and B, producing $\hat{i}_{k-frac-A}$, $\hat{i}_{k-A}$, $\hat{i}_{k-frac-B}$, and $\hat{i}_{k-B}$.

Interpolation is required to yield the same or nearly the same transmission times of the mixed P(Y) codes from the two receivers. Otherwise, a large loss of correlation power will occur due to the high chipping rate of the P(Y) code (10.23 MHz). In Ref. [8] a linear interpolation of the samples from one receiver is per-

formed such that the data from both receivers were at the same effective transmission times. This type of interpolation is avoided to minimize real-time computing requirements and a simple nearest neighbor interpolation is performed instead. This interpolation has the effect of choosing the set of samples from each receiver such that at the midpoint of the accumulation interval there is a maximum transmission time offset of $\frac{\delta T}{2}$ between the C/A codes from the two receivers. The effective code start times introduced in connection with Eqs. 4.10 and 4.11 are used to perform the sample matching needed for this interpolation.

To determine the first sample to use in sub-accumulation, the difference of the two effective fractional code start times is taken:

$$\Delta\hat{i}_{k-frac} = \hat{i}_{k-frac-A} - \hat{i}_{k-frac-B} \tag{4.12}$$

The index of the first sample for each sub-accumulation from each receiver is then:

$$\tilde{i}_{k-A} = \hat{i}_{k-A} + \max\left[\text{round}\left(\frac{\Delta\hat{i}_{k-frac}}{S_t}\right), 0\right] \tag{4.13a}$$

$$\tilde{i}_{k-B} = \hat{i}_{k-B} - \min\left[\text{round}\left(\frac{\Delta\hat{i}_{k-frac}}{S_t}\right), 0\right] \tag{4.13b}$$

This nearest-neighbor interpolation may cause a loss in correlation power due to residual P(Y) code mis-alignment between the two receivers. Another contribution of this work is calculation of an additional loss factor that accounts for any such reduction in cross-correlation power. This loss factor is denoted L_{pxc}, and its value depends on the shape of the autocorrelation function of the P(Y) code after it is filtered by the receivers' front-ends. It also depends on the magnitude of the code offset error during each sub-accumulation interval. This

error for each sub-accumulation is

$$\hat{i}_{err} = \begin{cases} \frac{\Delta\hat{i}_{k-frac}}{S_T} & \text{if } |\Delta\hat{i}_{k-frac}| \leq \frac{S_T}{2} \\ 1 - \frac{\Delta\hat{i}_{k-frac}}{S_T} & \text{if } \Delta\hat{i}_{k-frac} > \frac{S_T}{2} \\ 1 + \frac{\Delta\hat{i}_{k-frac}}{S_T} & \text{if } \Delta\hat{i}_{k-frac} < \frac{-S_T}{2} \end{cases} \tag{4.14}$$

and is measured in units of samples.

To determine the shape of the filtered P(Y) autocorrelation function, the RF front-end filter response function has been estimated using off-line system identification techniques as in Ref. [40]. Then, using this response and a simple triangular model for the un-filtered P(Y) code autocorrelation function, the real part of the filtered P(Y) code autocorrelation function has been calculated. Figure 4.2 plots both the original wide-band P(Y) code autocorrelation function (green dash-dotted line) and the filtered P(Y) code autocorrelation function (solid blue line).

In performing the cross-correlation required to detect the presence of spoofing, the instantaneous cross-correlation power loss equals the value of the filtered autocorrelation function of Fig. 4.2 evaluated at the offset between the P(Y) codes from the two receivers. Consistent with Eq. 4.14, the maximum code misalignment error is $\pm\frac{1}{2}$ of a front-end sample. This equals ± 0.895 P(Y) code chips. These maximum offsets are shown in Fig. 4.2 as the vertical dashed red lines. Therefore, the worst case power loss factor caused by nearest-neighbor interpolation is 0.87 (-0.6 dB). This small loss justifies the decision to use nearest neighbor interpolation in order to simplify the real-time signal processing.

If the spoofing detection statistic were calculated over a single C/A code period, the correlation power loss would be calculated by evaluating the fil-

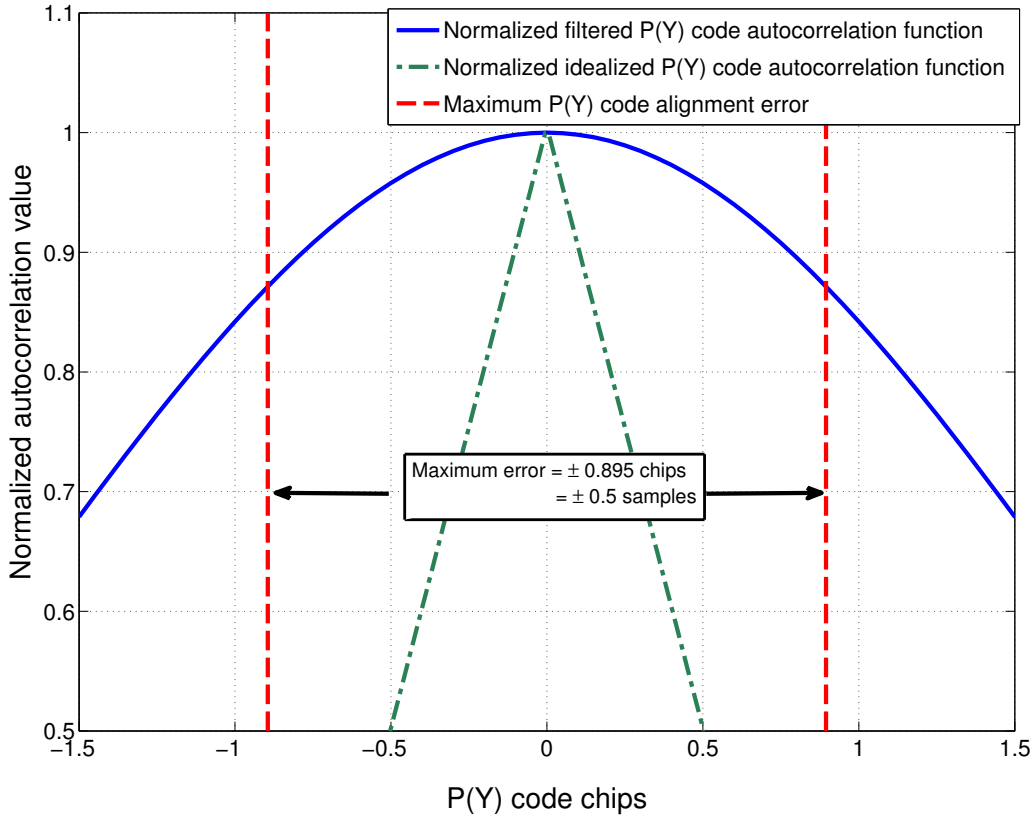

Figure 4.2: Filtered P(Y) code normalized autocorrelation function.

tered autocorrelation function at the code alignment error value $\hat{i}_{err}$. Because the spoofing detection statistic is calculated by summing over many C/A code periods, an alternate approach is used. The correlation power is computed for each individual period, and the results averaged over the full accumulation period of the detection statistic to produce L_{pxc}. In this implementation, the filtered autocorrelation function values are stored in a look-up table. This value L_{pxc} is used as part of the spoofing detection hypothesis calculations in the following subsection.

Once the correct sample index has been selected for the start of the sub-accumulation interval, carrier mixing must be performed. Note that at this point

in the processing, the receiver has already performed carrier mixing in order to compute the C/A code in-phase and quadrature accumulations, I_k and Q_k, as per Eq. 4.3. If this data that had been previously mixed with the quadrature carrier replica were buffered, no additional work would be necessary. The software would select the portion of this carrier-mixed data that begins at sample index $\tilde{i}_k$ from each receiver and use that in the cross-correlation. Rather than buffer the data, in this implementation the carrier replica is again mixed with the data, starting at the chosen index. This was a design decision made to allow more flexibility in the prototype system.

Given the PLL's carrier phase estimate $\hat{\phi}$, the start index of the sub-accumulation period $\tilde{i}_k$, and the carrier Doppler shift $\hat{\omega}_D$, a quadrature carrier replica is generated using a technique based on Ref. [43], and the quadrature part of the signal is isolated. For the k^{th} code period, the quadrature base-band mixed signal is

$$y_{qi} = y_i \sin\left[\omega_{IF} t_i + \hat{\phi}_k + \hat{\omega}_{Dk}\left(t_i - \tau\right)\right] \text{ for } t_i = \delta T[\tilde{i_k}, ..., (\tilde{i_k} + N - 1)] \tag{4.15}$$

This quantity is produced for each receiver, after which the un-normalized spoofing detection statistic is computed. This statistic is simply:

$$\gamma_u = \sum_{k=0}^{M-1} \sum_{\Delta i=0}^{N-1} y_{qA(\tilde{i}_{k-A}+\Delta i)} y_{qB(\tilde{i}_{k-B}+\Delta i)} \tag{4.16}$$

where M is the number of C/A code periods in the accumulation, and $y_{qA(\tilde{i}_{k-A}+\Delta i)}$ and $y_{qB(\tilde{i}_{k-B}+\Delta i)}$ are the quadrature base-band data samples at sample index $\tilde{i}_{k-A} + \Delta i$ and $\tilde{i}_{k-B} + \Delta i$ from receivers A and B, respectively. This implementation differs somewhat significantly from that in Ref. [8]. In that work,

the in-phase and quadrature accumulations I_k and Q_k were used to correct for any PLL tracking errors. That work made the assumption that noise effects on I_k and Q_k were negligible, which for this work proved not to be the case. Producing the detection statistic γ_u as prescribed in Ref. [8] led to a result with less power than simple multiplication by the quadrature carrier replica.

The actual implementation of Eqs. 4.15 and 4.16 is slightly different than the definitions of the quantities imply for reasons of computational efficiency. The receivers use 2-bit quantization and bitwise parallel operations [44]. This means that the data and the local carrier and code replicas are all quantized to two bits, with one bit indicating the sign of element, and the other bit indicating the magnitude. 32 consecutive elements are then stored in two 32-bit integers, with all of the sign bits stored in one integer, and all the magnitude bits stored in the other. This allows processing of 32 samples in parallel. Mixing of various elements (e.g., mixing the carrier replica with the data) involves only shifts and logical operations on the 32-bit integers. This approach has been exploited in order to design an efficient algorithm for performing cross-correlation between the two data streams.

First, each data stream is partially mixed with its respective carrier replica. This partial carrier mixing does two things: it shifts the data so that the first bit of the first 32-bit word in the accumulation interval corresponds to the desired sample index $\tilde{i}_{k-A}$ (or $\tilde{i}_{k-B}$), and it multiplies the sign bits of the carrier replica with the sign bits of the data. The results of this operation from the two receivers are then multiplied together. These multiplies are done via the exclusive-or operation. The rest of the cross-correlation is done with a look-up table, with the input being a 20-bit word composed of 4 bits of the multiplied carrier sign bits

and data sign bits, 4 data magnitude bits from each receiver, and 4 carrier replica magnitude bits from each receiver. The look-up table output is the integer that results from the multiply-and-accumulate of the 4 samples from each receiver and their respective carrier replicas. This result is accumulated for the desired integration time (i.e., a total of $M * N$ samples, from Eq. 4.16), and produces the un-normalized spoofing detection statistic γ_u. For a good description of similar bit-wise parallel software radio calculations, see Ref. [44].

Note that the bit-wise parallel sine and cosine replicas used in Eqns. 4.3 and 4.15 have amplitudes greater than one. The effect of these non-unit amplitudes divide out of the final normalized spoofing detection statistic presented in the next subsection.

4.5.2 Spoofing Detection Hypothesis Test

In addition to the un-normalized spoofing detection statistic γ_u, several other quantities must be calculated in order to implement a precise hypothesis test. Following directly from Ref. [8], two hypotheses are presented: H_0, the hypothesis that the receiver is free of spoofing, and H_1, the hypothesis that the receiver is being spoofed. The mean and variance of the spoofing detection statistic γ_u under H_1 are:

$$\bar{\gamma}_{u|H1} = 0 \tag{4.17}$$

$$\sigma^2_{\gamma u|H1} = \frac{MN}{4}\sigma^2_{RFA}\sigma^2_{RFB}\left[1 + 2\delta T(C/N_0)_{pyA}\right] \tag{4.18}$$

The mean value under this hypothesis is zero because if the receiver is be-

ing spoofed, the effective received P(Y) code power is zero, thus the cross-correlation produces no power.

Under H_0, the spoofing detection statistic mean and variance are:

$$\bar{\gamma}_{u|H0} = MN\sigma_{RFA}\sigma_{RFB}\delta T L_{pxc}\sqrt{(C/N_0)_{pyA}(C/N_0)_{pyB}} \tag{4.19}$$

$$\sigma^2_{\gamma u|H0} = \frac{MN}{4}\sigma^2_{RFA}\sigma^2_{RFB}\{1 + 2\delta T[(C/N_0)_{pyA} + (C/N_0)_{pyB})]\} \tag{4.20}$$

where L_{pxc} is the cross-correlation loss factor introduced in the previous subsection.

These terms are then normalized by the standard deviation under the hypothesis of spoofing, H_1, resulting in these means and standard deviations:

$$\bar{\gamma}_{norm|H1} = 0 \tag{4.21}$$

$$\sigma_{\gamma norm|H1} = 1 \tag{4.22}$$

$$\bar{\gamma}_{norm|H0} = 2\delta T L_{pxc}\sqrt{\frac{MN(C/N_0)_{pyA}(C/N_0)_{pyB}}{1 + 2\delta T(C/N_0)_{pyA}}} \tag{4.23}$$

$$\sigma_{\gamma norm|H0} = \sqrt{\frac{1 + 2\delta T[(C/N_0)_{pyA} + (C/N_0)_{pyB})]}{1 + 2\delta T(C/N_0)_{pyA}}} \tag{4.24}$$

The same normalization must be applied to the spoofing detection statistic that has been calculated from the quadrature samples:

$$\gamma_{norm} = \frac{\gamma_u}{\sigma_{RFA}\sigma_{RFB}\sqrt{\frac{MN}{4}\left[1 + 2\delta T(C/N_0)_{pyA}\right]}} \tag{4.25}$$

Assuming the probability density functions under the spoofed and unspoofed hypotheses, $p(\gamma_{norm}|H_1)$ and $p(\gamma_{norm}|H_0)$, are Gaussian [8], and given a probability of false alarm α_{fa}, a spoofing detection threshold γ_{th} can be computed by solving the following equation:

$$\begin{aligned}\alpha_{fa} &= \int_{-\infty}^{\gamma_{th}} p(\gamma_{norm}|H_0)d\gamma_{norm} \\ &= \frac{1}{\sqrt{2\pi}\sigma_{\gamma norm|H0}} \int_{-\infty}^{\gamma_{th}} \exp\left[-\frac{(\gamma_{norm}-\bar{\gamma}_{norm|H0})^2}{2\sigma^2_{\gamma norm|H0}}\right] d\gamma_{norm}\end{aligned} \tag{4.26a}$$

In practice, this equation is solved off-line by assuming a zero-mean, unit-variance distribution and choosing a small value of α_{fa} ($\alpha_{fa} = 0.0001$ for all results herein), and the result is embedded in the source code. During processing, this result is then multiplied by $\sigma_{\gamma norm|H0}$ and then added to $\bar{\gamma}_{norm}$ to produce the threshold γ_{th} required for hypothesis testing. The actual output of the receiver is

$$\tilde{\gamma} = \gamma_{norm} - \gamma_{th} \tag{4.27}$$

for each of the satellites that are common to both receivers A and B. If $\tilde{\gamma} > 0$ for a particular satellite, it is determined that that particular signal is not being spoofed. If $\tilde{\gamma} \leq 0$, that channel is marked as being spoofed.

The probability of successfully detecting a spoofing attack is

$$\begin{aligned}P_d &= \int_{-\infty}^{\gamma_{th}} P(\gamma_n|H_1)\,\mathrm{d}\gamma_n \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\gamma_{th}} \exp(-0.5\gamma^2_{norm})d\gamma_{norm}\end{aligned} \tag{4.28a}$$

Note that P_d depends on the spoofing detection statistic threshold, which itself depends on the the integration time, on the noise variance, and on the signal carrier-to-noise ratio. The latter quantity varies with time due to environmental

changes such as satellite elevation. Rather than aiming for a fixed P_d and adjusting the integration time to account for variations in carrier-to-noise ratio, a fixed integration time has been chosen for all signals, regardless of their carrier-to-noise ratio. For this implementation, a nominal integration time of 2 seconds has been used. For a C/A code carrier-to-noise ratio of 50 dB-Hz at both the reference and defended receivers, for a P(Y) power transmission decrement of L_{psv} of 3 dB, for a false alarm probability of 0.01%, and for a 2 second integration time, P_d is greater than 99.999%.

Although P_d varies for each signal, α_{fa} is always fixed and small, so any indication of spoofing is a reliable indicator that that signal is being spoofed.

4.6 Results

Several tests have been conducted using this implementation of codeless spoofing detection. For these tests, a receiver located on the roof of a building in Ithaca, New York (42.44° N, 76.48° W) was used as the reference receiver, and a receiver located on the roof of a building in Austin, Texas (30.29° N, 97.74° W) was used as the defended receiver.

4.6.1 Determining The Per-Satellite Power Variation

Initial results showed some disagreement between the expected value of the spoofing test statistic and its actual value, which varied by satellite. It was theorized that, for some satellites, the difference in power between the transmitted C/A code signal and the transmitted P(Y) code signal varied from the

3 dB decrement suggested by Ref. [37]. To explore this possibility, data were collected from both receivers every half hour on February 7, 8, and 12, 2013. It was assumed that both receivers were free of spoofing during this period (and indeed, none was detected). It was also assumed that the actual transmission power decrement L_{psv} was exactly 3 dB, and a correction to this value was solved for. These data were processed using the above algorithms, with $\bar{\gamma}_u$ and γ_u calculated over two second accumulation intervals for the length of each data set, which was nominally 150 seconds. The mean values of $\bar{\gamma}_u$ and γ_u over the length of the data set were calculated, and the corrected L_{psv} is solved for:

$$L_{psv} = -3 + 10log_{10}\left(\frac{1}{P}\sum_{j=1}^{P}\frac{E[\gamma_{uj}]}{E[\bar{\gamma}_{uj}]}\right) \tag{4.29}$$

where the -3 in this equation is the assumed difference in transmission power in dB [37] and P is the number of data sets used in calculating L_{psv} for each satellite. The average was $P = 12.2$, with the fewest number of data sets being 9 for PRN 29. The resulting L_{psv} values are presented in Table 4.1 for every satellite in the GPS constellation as of February, 2013. Also presented are their standard deviations.

The smallest power decrement was -2.32 dB for PRN 3, and the largest power decrement is -2.93 dB for PRN 4. These results also illustrate that the algorithm is operating correctly in the absence of spoofing, as the un-normalized spoofing detection statistic closely matches the mean value predicted from the C/A code carrier-to-noise ratio after applying this small correction. Had the corrected L_{psv} values been wildly different from -3 dB, one might have questioned the validity of the analysis on which this entire spoofing detection technique is based.

The values presented in Table 4.1 are subject to change for a variety of reasons. These include the addition or removal of satellites into the active GPS constellation, varying P(Y) code transmission power ("flex power" [37]), or switching between different transmitters on the satellite. These values should be monitored continuously and updated as needed.

4.6.2 Detection of a Spoofing Attack

To test the capability of this system to detect spoofing, a sophisticated GPS spoofer [4, 5, 45, 46] has been used to conduct a number of attacks in a controlled environment. For each test, the output of the GPS spoofer was combined electrically with the signal coming from the roof-mounted antenna. This was done to ensure that no harmful signals were transmitted over the air.

Consider a representative test that was conducted on July 14, 2011, with the spoofer co-located with the defended receiver in Austin. Fig. 4.3 illustrates this attack, with a signal that is being spoofed shown in the top panel, and an unspoofed signal shown in the bottom panel. For both figures, the normalized spoofing detection statistic, γ_{norm} is the solid blue line, its expected value $\bar{\gamma}_{norm}$ is the dash-dotted green line, and the spoofing detection threshold, γ_{th} is the dashed red line.

During this test, there was no spoofing of any signal for the first 60 seconds, to establish a baseline for normal operation. At 60 seconds, spoofing of some of the signals began, but the spoofer transmitted its best estimate of the "true" signal, though it did not maintain phase coherence with the true signal. For the spoofer to capture the target receiver, the transmitted power must be larger

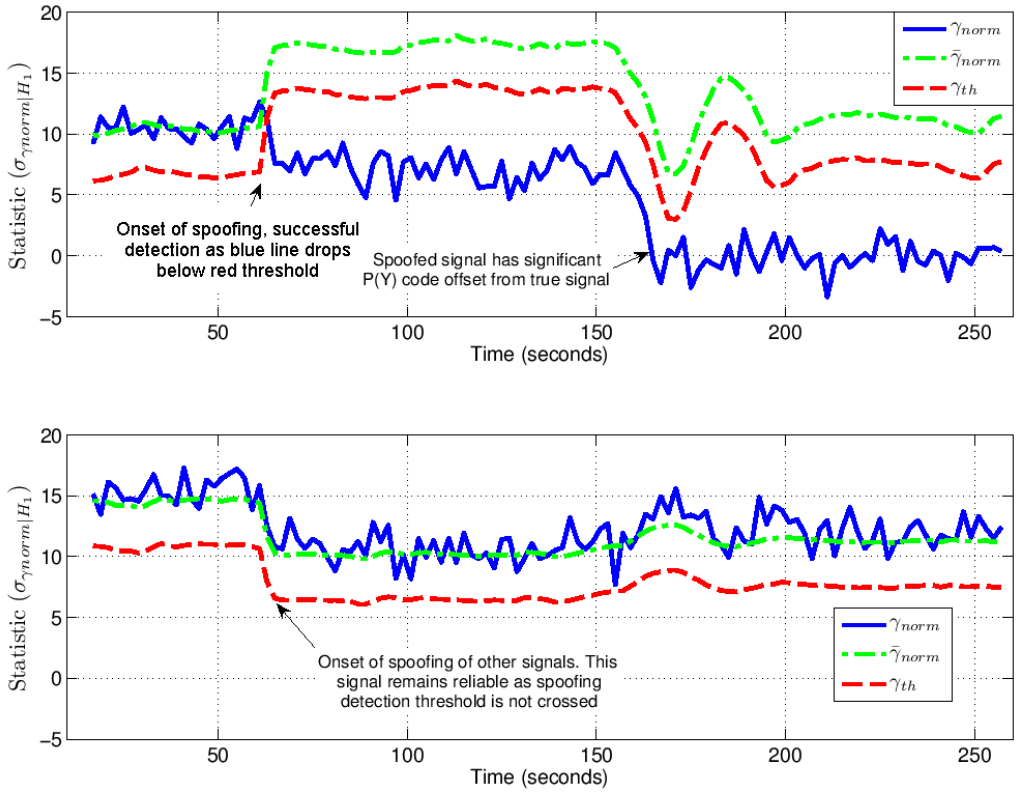

Figure 4.3: Normalized spoofing detection statistic and related quantities for a spoofed signal (top panel) and an un-spoofed signal (bottom panel) during a spoofing attack.

than the power received from the satellite. This increase in power has the effect of reducing the gain in the receiver due to the action of its automatic gain control. For the spoofed signal, this reduction in gain causes the P(Y) code cross-correlation to fall. The received C/A code power, on the other hand, rises due to the increased power of the spoofer. This rise causes $\bar{\gamma}_{norm}$ and γ_{th} to rise. These two effects combine to cause γ_{norm} to cross below the spoofing detection threshold, triggering an alarm. Thus, even though the spoofer was not "lying" to the receiver about its location, it was still detected by virtue of its higher C/A code power level and the lower power level of the P(Y) code.

For the un-spoofed signal shown in the bottom panel, the reduced gain acted

equally on both the C/A and P(Y) code signal. Thus the cross-correlation detection statistic's expected value remained accurate and no spoofing was detected.

Approximately 90 seconds later, at receiver time 150 seconds, the spoofer began moving the spoofed signal away from the true signal, and the spoofing detection statistic for the spoofed signal became zero-mean, as would be expected once the spoofed C/A code drags the defended receiver timing far away from the true P(Y) code.

The actual spoofing detection output for all signals, $\tilde{\gamma}$ from Eq. 4.27, during this same spoofing attack is shown in Fig. 4.4. It is quite clear that PRNs 23 and 30 remained free of spoofing, while all the other signals were spoofed from 60 seconds onward. These results are consistent with the spoofing test parameters.

The probability of successful spoofing detection for all signals during this spoofing attack is shown in Fig. 4.5. This probability is quite high for most of the signals for the duration of the test. PRN 30 had a significantly lower detection probability after the onset of the spoofing attack at 60 seconds. After this time, the carrier-to-noise ratio of the C/A code for this PRN was quite low, approximately 38 dB-Hz, which is the cause of this low detection probability. This "problem" occurs because PRN 30 is not being spoofed, which gives it a lower carrier-to-noise ratio. Had it been spoofed with a signal strong enough to drag off the receiver tracking loops, the problem would not have occurred, and the actual spoofing attack would likely have been detected. The drop in P_d seen on several signals at approximately 160 seconds is due to a rapid drop in the carrier-to-noise ratio of those signals, presumably due to the spoofed and true C/A codes briefly interfering with each other as the spoofer drags the receiver away from the true signal.

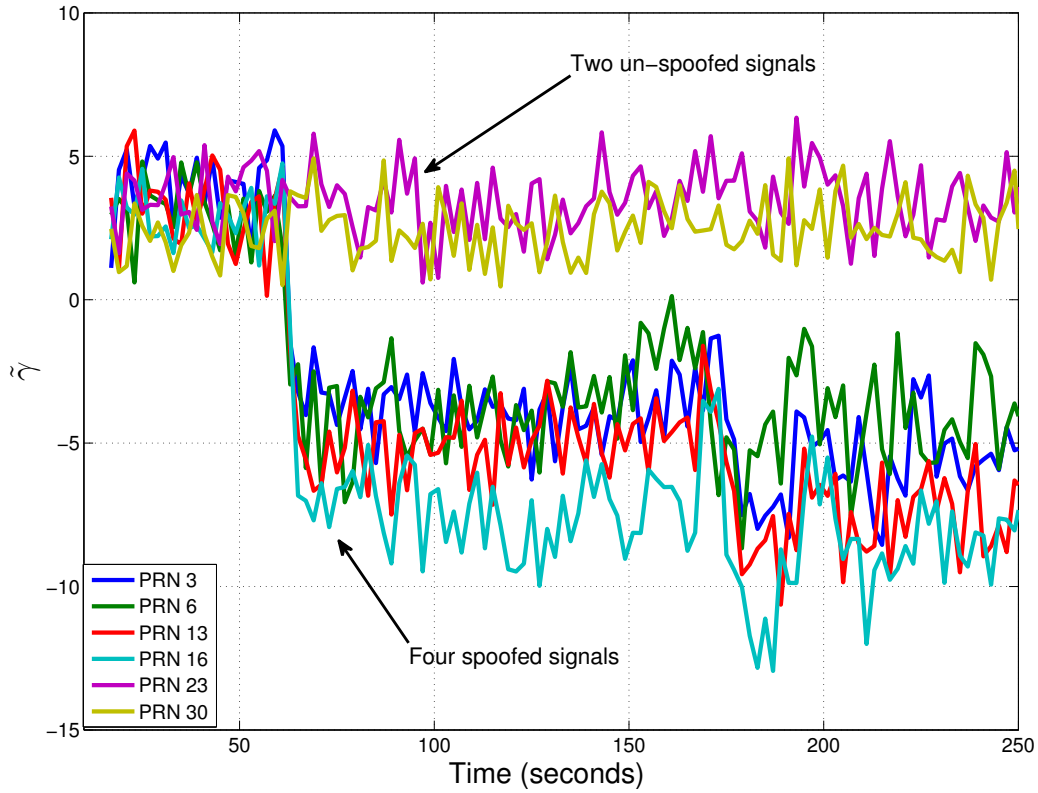

Figure 4.4: Receiver output for all signals during a spoofing attack.

These results and many similar tests represent the first detections of spoofing attacks in a real-time system using a single antenna per receiver. This also represent the first real-time implementation of the cryptographic spoofing detection technique of Refs. [32, 33, 34, 8].

4.7 Summary and Conclusion

A method for detecting spoofing of the GPS L1 C/A code signal has been implemented in a real-time system. This method assumes that the encrypted P(Y) code signal is free of spoofing. This assumption allows the use of a "reference

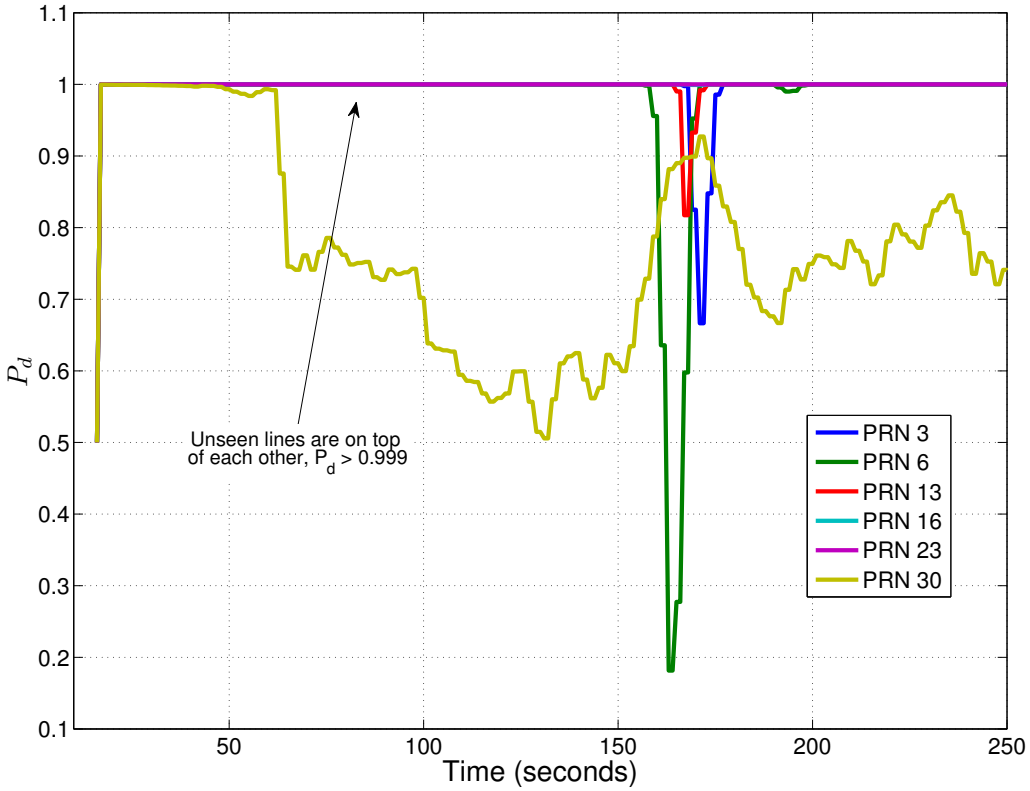

Figure 4.5: Probability of detection for all signals during a spoofing attack.

receiver" that is free of spoofing to assist in detection of possible spoofing at a "defended receiver". If only the C/A code is being spoofed, it is possible to do a cross-correlation with the portion of the data from the two receivers that is in quadrature with the C/A code. The code and carrier phase relationship between the C/A and P(Y) code signals are known, and the portion of the signal containing the P(Y) code can be isolated and used for cross-correlation. If there is no spoofing at the defended receiver, this cross-correlation will result in a large spoofing detection statistic due to the P(Y) code autocorrelation properties. If the defended receiver is being spoofed, this has the effect of introducing a code phase offset between the spoofed C/A code and the un-spoofed P(Y) code, resulting in a small spoofing detection statistic.

A hypothesis test is constructed to allow determination of a threshold for the spoofing detection statistic. This threshold guarantees a low probability of false alarm. For sufficiently strong signals and sufficiently long detection intervals, the probability of detection is very close to one.

The new method has been tested by subjecting it to realistic spoofing attacks. Successful detection of these spoofing attacks has been demonstrated. Nominal receiver response in the absence of spoofing has also been demonstrated in a number of tests.

As a side benefit of this work, the power differences between the C/A and P(Y) code signals has been investigated experimentally. The actual differences in power levels between these two signals can vary by more than a decibel from the value implied in the system documentation. These power difference calibrations have been used to develop the precise hypothesis test analysis on which the new spoofing detection method is based.

PRN	Mean (dB)	σ (dB)	PRN	Mean (dB)	σ (dB)
1	-2.87	0.04	17	-2.92	0.03
2	-2.92	0.04	18	-2.88	0.05
3	-2.32	0.13	19	-2.83	0.07
4	-2.93	0.05	20	-2.82	0.06
5	-2.92	0.03	21	-2.91	0.08
6	-2.86	0.05	22	-2.91	0.03
7	-2.90	0.06	23	-2.89	0.05
8	-2.90	0.04	24	-2.80	0.08
9	-2.83	0.05	25	-2.88	0.06
10	-2.87	0.06	26	-2.78	0.14
11	-2.89	0.08	27	-2.84	0.05
12	-2.87	0.04	28	-2.61	0.02
13	-2.88	0.04	29	-2.89	0.03
14	-2.89	0.03	30	-2.88	0.05
15	-2.89	0.03	31	-2.89	0.03
16	-2.92	0.03	32	-2.83	0.04

Table 4.1: Observed decrement in transmitted power between L1 C/A and L1 P(Y) signals, by satellite.

CHAPTER 5

SUMMARY AND CONCLUSIONS

A software-defined GPS receiver called CASES was built that implements several novel signal processing techniques. This receiver has been specially designed to be robust to ionospheric scintillation and to provide measurements that are useful for the study of ionospheric phenomena. This receiver was implemented on a custom-designed DSP board paired with a narrow-bandwidth dual-frequency radio front-end. The receiver implements a differencing technique to remove errors due to oscillator effects, an advanced triggering mechanism to detect the onset of signal scintillation, data bit prediction and wipe-off for more robust signal tracking, and data buffering to allow observation of the onset of scintillation. Receiver performance was verified using both real scintillation observed during field campaigns and simulated RF data containing scintillation based on an empirical model.

In the course of using this receiver for ionospheric study, a very subtle fault was discovered in the carrier phase of the signal transmitted by GPS satellite SVN 48. The details of this signal fault are shown, along with verification of the discovery using measurements from a commercial GPS receiver. Several other satellites of the same design (i.e., GPS Block IIR-M) were investigated and no similar faults were observed. The implications of this fault were discussed.

A GPS spoofing detection algorithm that makes use of the unknown but presumably secure GPS P(Y) code was implemented using a version of CASES that runs on a personal computer. This technique makes use of data collected simultaneously from two locations, one of which is assumed to be free of spoofing. Cross-correlation of the portion of the data from both locations that should con-

tain the unknown P(Y) code is done, and a hypothesis test is implemented to determine if the result indicates the presence of spoofing. This technique executes in real time (albeit with a negligible delay) to quickly determine the presence of spoofing. This technique was tested using RF signal spoofing at one of the locations, and spoofing was successfully detected. Additionally, previously unpublished decrements in transmitted signal power between the L1 C/A signal and the L1 P(Y) signals as of 2013 are provided for each satellite.

BIBLIOGRAPHY

[1] Anon., "IS-GPS-200E: Navstar GPS space segment/navigation user interfaces," tech. rep., Science Applications International Corporation, 2010. http://www.losangeles.af.mil/library/factsheets/factsheet.asp?id=9364.

[2] Anon., "Vulnerability assessment of the transportation infrastructure relying on the Global Positioning System," tech. rep., John A. Volpe National Transportation Systems Center, 2001.

[3] J. S. Warner and R. G. Johnston, "A simple demonstration that the Global Positioning System (GPS) is vulnerable to spoofing," *Journal of Security Administration*, 2003.

[4] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, Jr., "Assessing the spoofing threat: development of a portable GPS civilian spoofer," in *Proceedings of the ION GNSS Meeting*, (Savannah, GA), Institute of Navigation, 2008.

[5] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, Jr., "Assessing the spoofing threat," *GPS World*, vol. 20, pp. 28–38, Jan. 2009.

[6] S. Lo, D. DeLorenzo, P. Enge, D. Akos, and P. Bradley, "Signal authentication," *Inside GNSS*, vol. 0, pp. 30–39, Sept. 2009.

[7] M. L. Psiaki, B. W. O'Hanlon, J. A. Bhatti, and T. E. Humphreys, "Civilian GPS spoofing detection based on dual-receiver correlation of military signals," in *Proceedings of the ION GNSS Meeting*, (Portland, OR), Sept. 2011.

[8] M. Psiaki, B. W. O'Hanlon, J. Bhatti, and T. E. Humphreys, "GPS spoofing detection via dual-receiver correlation of military signals," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, pp. 2250–2267, Oct. 2013.

[9] L. Wanninger, "Effects of the equatorial ionosphere on GPS," *GPS World*, vol. 4, pp. 48–52, July 1993.

[10] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, and P. M. Kintner, Jr., "GNSS receiver implementation on a DSP: Status, challenges, and prospects," in *Proceedings of the ION GNSS Meeting*, (Fort Worth, TX), Institute of Navigation, 2006.

[11] B. W. O'Hanlon, M. L. Psiaki, P. M. Kintner, Jr., and T. E. Humphreys, "Development and field testing of a DSP-based dual-frequency software GPS receiver," in *Proceedings of the ION GNSS Meeting*, (Savannah, GA), Institute of Navigation, 2009.

[12] A. J. Van Dierendonck, "How GPS receivers measure (or should measure) ionospheric scintillation and TEC and how GPS receivers are affected by the ionosphere," in *Proc. 11th International Ionospheric Effects Symposium*, (Alexandria, VA), 2005.

[13] T. E. Humphreys, M. L. Psiaki, B. M. Ledvina, and P. M. Kintner, Jr., "GPS carrier tracking loop performance in the presence of ionospheric scintillations," in *Proceedings of the ION GNSS Meeting*, (Long Beach, CA), Institute of Navigation, Sept. 2005.

[14] S. Skone, K. Knudsen, and M. de Jong, "Limitations in GPS receiver tracking performance under ionospheric scintillation conditions," *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy*, vol. 26, no. 68, pp. 613 – 621, 2001. Proceedings of the First COST Action 716 Workshop Towards Operational GPS Meteorology and the Second Network Workshop of the International GPS Service (IGS).

[15] C. L. Rino, V. H. Gonzalez, and A. R. Hessing, "Coherence bandwidth loss in transionospheric radio propagation," *Radio Science*, vol. 16, pp. 245–255, 1981.

[16] E. J. Fremouw, R. L. Leadabrand, R. C. Livingston, M. D. Cousins, C. L. Rino, B. C. Fair, and R. A. Long, "Early results from the DNA Wideband Satellite experiment - Complex-signal scintillation," *Radio Science*, vol. 13, pp. 167–187, Feb. 1978.

[17] A. J. Van Dierendonck, J. Klobuchar, and Q. Hua, "Ionospheric scintillation monitoring using commercial single frequency C/A code receivers," in *Proceedings of the ION GPS Meeting*, (Portland, Oregon), pp. 1333–1342, Institute of Navigation, 1993.

[18] T. E. Humphreys, M. L. Psiaki, B. M. Ledvina, A. P. Cerruti, and P. M. Kintner, Jr., "A data-driven testbed for evaluating GPS carrier tracking loops in ionospheric scintillation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, pp. 1609–1623, Oct. 2010.

[19] T. Beach, "Perils of the GPS phase scintillation index (σ_ϕ)," *Radio Science*, vol. 41, 2006.

[20] T. E. Humphreys, M. L. Psiaki, and P. M. Kintner, Jr., "Modeling the effects of ionospheric scintillation on GPS carrier phase tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, pp. 1624–1637, Oct. 2010.

[21] M. K. Simon and W. Lindsey, "Optimum performance of suppressed carrier receivers with costas loop tracking," *IEEE Transactions on Communications*, vol. 25, pp. 215–227, February 1977.

[22] T. E. Humphreys, M. L. Psiaki, J. C. Hinks, B. O'Hanlon, and P. M. Kintner, Jr., "Simulating ionosphere-induced scintillation for testing GPS receiver phase tracking loops," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, pp. 707–715, Aug. 2009.

[23] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*. Lincoln, Massachusetts: Ganga-Jamuna Press, 2006.

[24] A. J. Van Dierendonck, *Global Positioning System: Theory and Applications*, ch. 8: GPS Receivers, pp. 329–407. Washington, D.C.: American Institute of Aeronautics and Astronautics, 1996.

[25] D. Shepard, J. Bhatti, and T. Humphreys, "Drone hack: Spoofing attack demonstration on a civilian unmanned aerial vehicle," *GPS World*, vol. 23, pp. 30–33, Aug. 2012.

[26] L. Scott, "Anti-spoofing and authenticated signal architectures for civil navigation systems," in *Proceedings of the ION GNSS Meeting*, (Portland, Oregon), pp. 1542–1552, Institute of Navigation, 2003.

[27] S. Peterson, "Iran hijacked US drone, says Iranian engineer," 2011. http://www.csmonitor.com/World/Middle-East/2011/1215/Exclusive-Iran-hijacked-US-drone-says-Iranian-engineer-Video.

[28] J. S. Warner and R. G. Johnston, "GPS spoofing countermeasures," Dec. 2003. `http://www.homelandsecurity.org/bulletin/DualBenefit/warner_gps_spoofing.html`.

[29] N. White, P. Maybeck, and S. DeVilbiss, "Detection of interference/jamming and spoofing in a DGPS-aided inertial system," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 34, no. 4, pp. 1208–1217, 1998.

[30] P. Y. Montgomergy, T. E. Humphreys, and B. M. Ledvina, "Receiver-

autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil GPS spoofer," in *Proceedings of the ION ITM*, (Anaheim, CA), Jan. 2009.

[31] M. L. Psiaki, S. P. Powell, and B. W. O'Hanlon, "GNSS spoofing detection by correlating carrier phase with rapid antenna motion," *GPS World*, vol. 24, pp. 53–58, June 2013.

[32] P. Levin, D. De Lorenzo, P. Enge, and S. Lo, "Authenticating a signal based on an unknown component thereof," June 2011. US Patent 7,969,354 B2.

[33] B. O'Hanlon, J. Bhatti, T. E. Humphreys, and M. Psiaki, "Real-time spoofing detection using correlation between two civil GPS receiver," in *Proceedings of the ION GNSS Meeting*, (Nashville, Tennessee), pp. 3584–3590, Institute of Navigation, 2012.

[34] M. L. Psiaki, B. W. O'Hanlon, J. A. Bhatti, and T. E. Humphreys, "Civilian GPS spoofing detection based on dual-receiver correlation of military signals," in *Proceedings of the ION GNSS Meeting*, (Portland, Oregon), pp. 2619–2645, Institute of Navigation, 2011.

[35] K. Wesson, M. Rothlisberger, and T. E. Humphreys, "Practical cryptographic civil GPS signal authentication," *NAVIGATION, Journal of the Institute of Navigation*, vol. 59, no. 3, pp. 177–193, 2012.

[36] T. E. Humphreys, "Detection strategy for cryptographic GNSS anti-spoofing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, pp. 1073–1090, April 2013.

[37] Anon., "IS-GPS-200G: Navstar GPS space segment/navigation user interfaces," tech. rep., Science Applications International Corporation, 2012. http://www.gps.gov/technical/icwg/IS-GPS-200G.pdf.

[38] C. Edgar, J. Price, and D. Iteigh, "GPS block IIA and IIR received signal power measurements," in *Proceedings of the ION NTM*, (Long Beach, CA), pp. 401–411, Institute of Navigation, Jan. 1998.

[39] C. Edgar, D. B. Goldstein, and P. Bently, "Current constellation GPS satellite ground received signal power measurements," in *Proceedings of the ION NTM*, (San Diego, CA), pp. 948–954, Institute of Navigation, Jan. 2002.

[40] M. L. Psiaki and B. W. O'Hanlon, "System identification of a GNSS re-

ceiver's RF filter impulse response function," in *Proceedings of the ION GNSS Meeting*, (Portland, Oregon), pp. 3690–3708, Institute of Navigation, 2011.

[41] Anon., "Recommendation for key management—Part I: General (revision 3)," sp 800-57, National Institute of Standards and Technology, July 2007.

[42] B. O'Hanlon, M. Psiaki, S. Powell, J. Bhatti, T. E. Humphreys, G. Crowley, and G. Bust, "CASES: A smart, compact GPS software receiver for space weather monitoring," in *Proceedings of the ION GNSS Meeting*, (Portland, Oregon), Institute of Navigation, 2011.

[43] B. M. Ledvina, "Efficient real-time generation of bit-wise parallel representations of oversampled carrier replicas," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, Oct. 2011.

[44] B. M. Ledvina, M. L. Psiaki, S. P. Powell, and P. M. Kintner, Jr., "Bit-wise parallel algorithms for efficient software correlation applied to a GPS software receiver," *IEEE Transactions on Wireless Communications*, vol. 3, Sept. 2004.

[45] T. E. Humphreys, J. A. Bhatti, D. P. Shepard, and K. D. Wesson, "The Texas spoofing test battery: Toward a standard for evaluating GNSS signal authentication techniques," in *Proceedings of the ION GNSS Meeting*, (Nashville, Tennessee), Institute of Navigation, 2012.

[46] D. P. Shepard, T. E. Humphreys, and A. A. Fansler, "Evaluation of the vulnerability of phasor measurement units to GPS spoofing attacks," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 3-4, pp. 146–153, 2012.